Final STAT547 Spring2014
On the Relation between Temperature and Precipitation from 73 Spanish Weather Stations

Yet Nguyen

May 2, 2014

1 Introduction

In this project, we will analyze the dataset `aemet` which is available in the package `fda.usc` (Functional Data Analysis and Utilities for Statistical Computing) developed by Manuel Oviedo de la Fuente et al. . The dataset is the series of daily summaries of 73 Spanish weather stations selected for the period 1980-2009. The dataset contains geographic information of each station and the average for the period 1980-2009 of daily temperature, daily precipitation and daily wind speed. The data come originally from Meteorological State Agency of Spain (AEMET) (`http://www.aemet.es/`).

Precipitation is the amount of watter that falls down from clouds. Previous research suggests that the temperature substantially affect precipitation. For instance, according to wikipedia: "during the Last Glacial Maximum of 18,000 years ago, thermal-driven evaporation from the oceans onto continental landmasses was low, causing large areas of extreme desert, including polar deserts (cold but with low rates of precipitation). In contrast, the world's climate was wetter than today near the start of the warm Atlantic Period of 8000 years ago."

In this project, we will look at the relationship between precipitation and temperature under the functional data analysis point of view. In particular, we will investigate the functional structure of those two quantities over daily measures for the period 1980-2009 by principal component analysis. And then we will do a functional linear models in the two forms. The first one is scale response precipitation (calculated as average precipitation over time) vs. temperature curves. The second one is functional response precipitation vs. functional predictor temperature.

2 Scatter plot of the two functional datasets

First of all, we can look at the two datasets and their relation which is shown in Figures 1 and 2

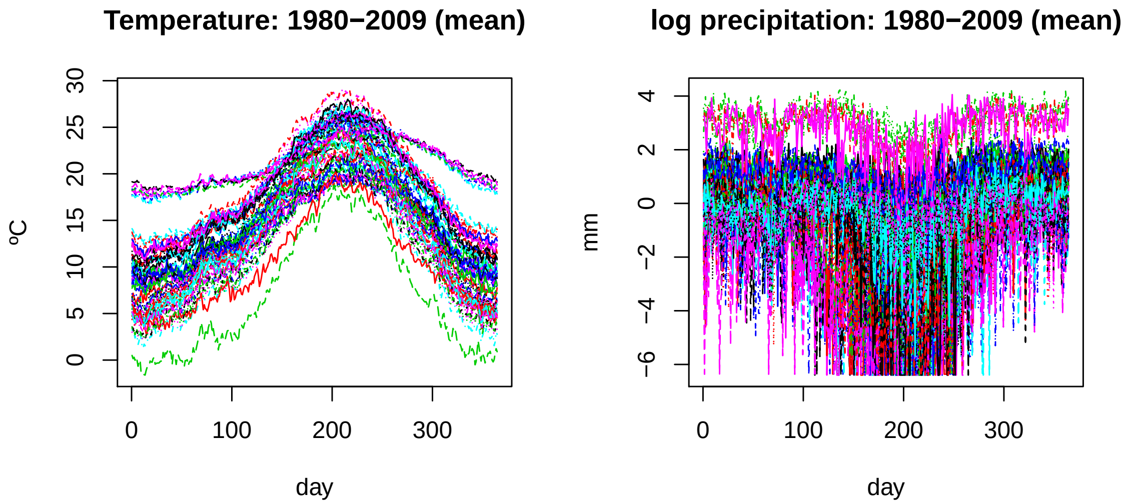

Figure 1: Scatter plot of temperature and log precipitation data

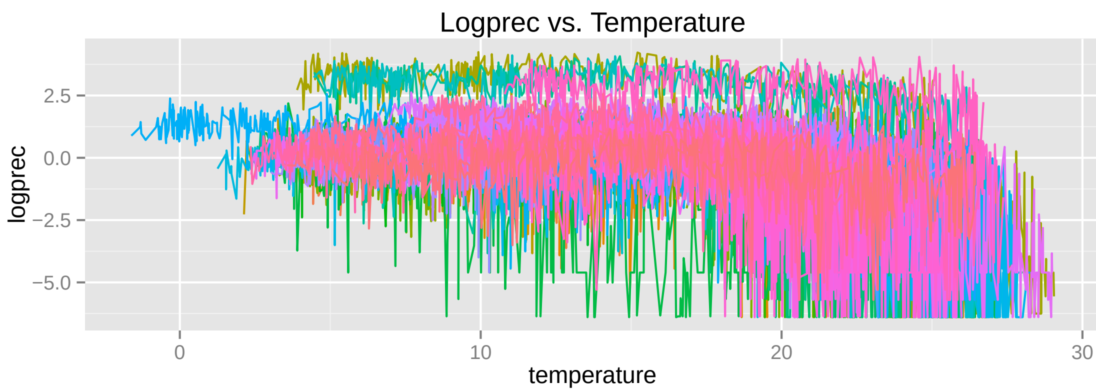

Figure 2: Relationship between temperature and precipitation

3 Covariance modelling and Principle Component Analysis for Temperature and Repricipitation

73 curves of each dataset have 365 time points, therefore, we can consider those as *dense* functional data, hence we will use Ramsay and Silverman's Approach to obtain the estimated mean and variance matrix of each data set.

3.1 Analysis for Temperature

The estimated mean and covariance is shown in Figure 3.

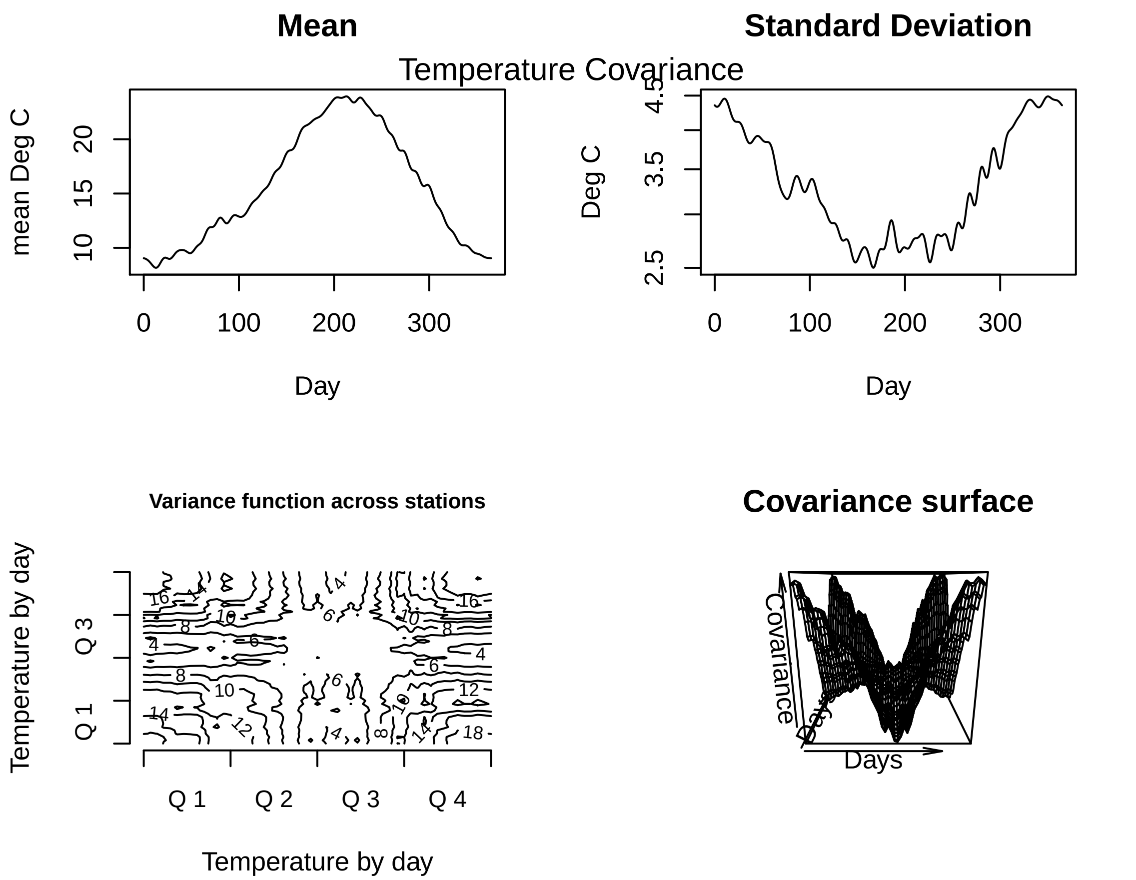

Figure 3: Estimated Mean and Covariance for Temperature data.

The estimated eigenvalues and eigenfunctions of temperature data are shown in Figure 4.

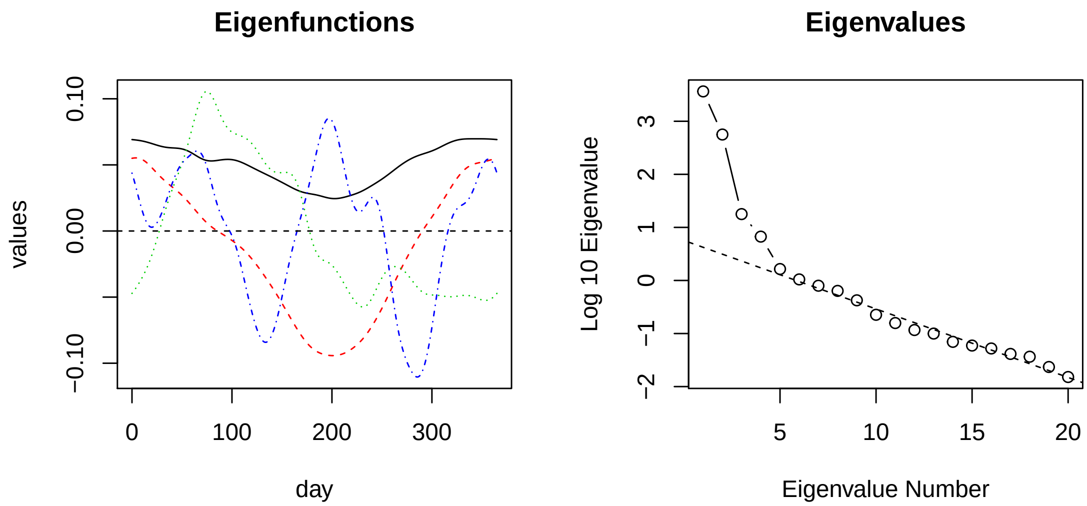

Figure 4: Estimated eigenvalues and eigenfunctions of temperature data.

3.2 Analysis for Precipitation

The estimated mean and covariance is shown in Figure 5.

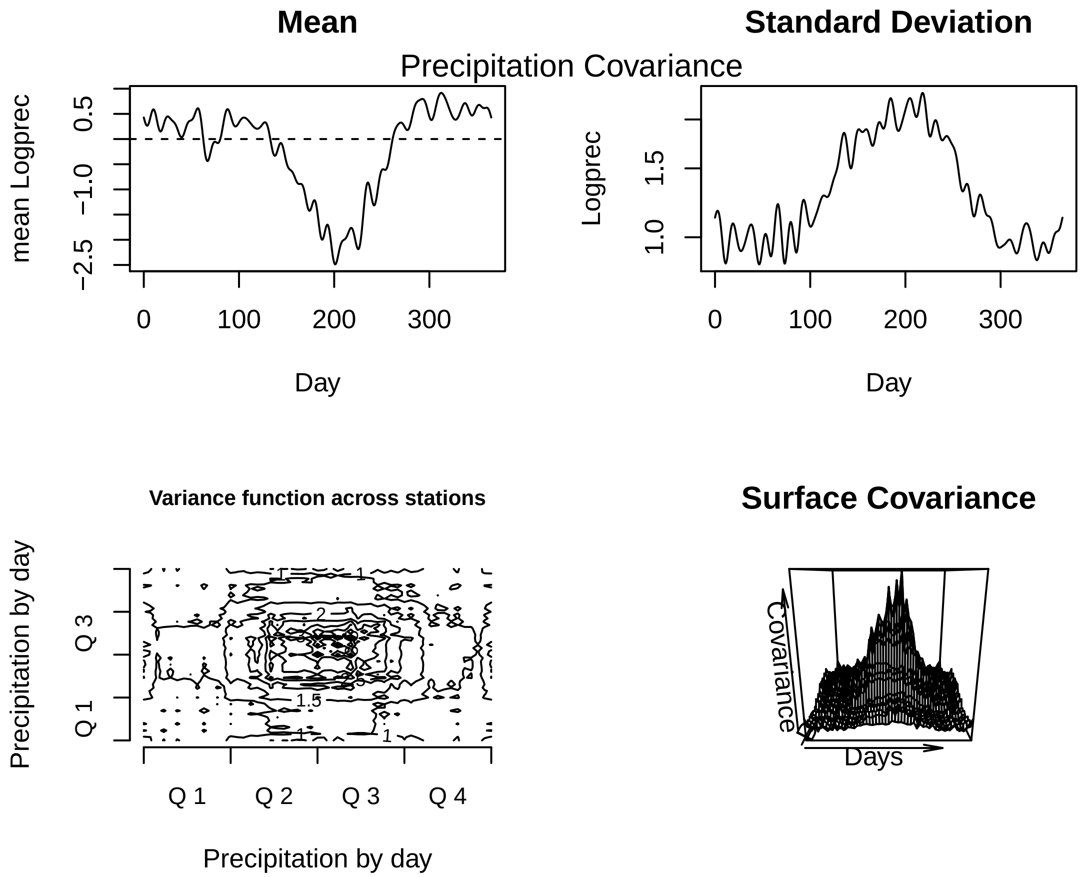

Figure 5: Estimated Mean and Covariance for Precipitation data.

The estimated eigenvalues and eigenfunctions of precipitation data are shown in Figure 6.

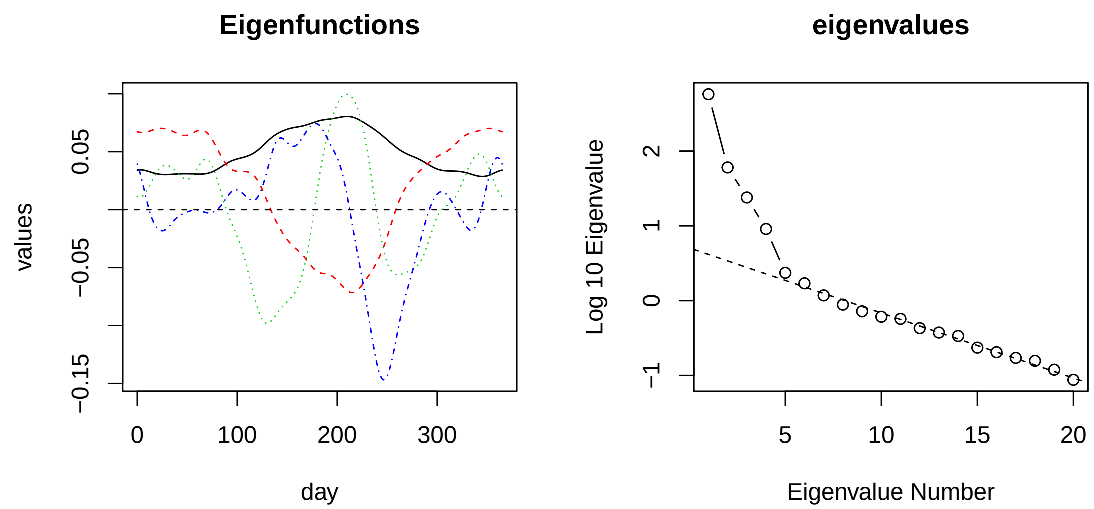

Figure 6: Estimated eigenvalues and eigenfunctions of precipitation data.

Looking at the Figures 3, 4, 5, and 6, we can see that the behavior of *temperature* and *precipitation* are differently in the opposite direction, i.e., somehow "negative" correlation, both in the average mean curve and covariance surface.These conclusions do support the hypotheses that there is negative correlation between temperature and precipitation. In the next sections, we will investigate the functional regression relation between these two functional datasets.

4 Functional Linear Model with Functional Covariate as Temperature and Scalar Response as Log of Total Precipitation:Ramsay and Silverman's Approach and Principle Component Analysis Approach

Figure 7 shows the scatter plot of log of total precipitation.

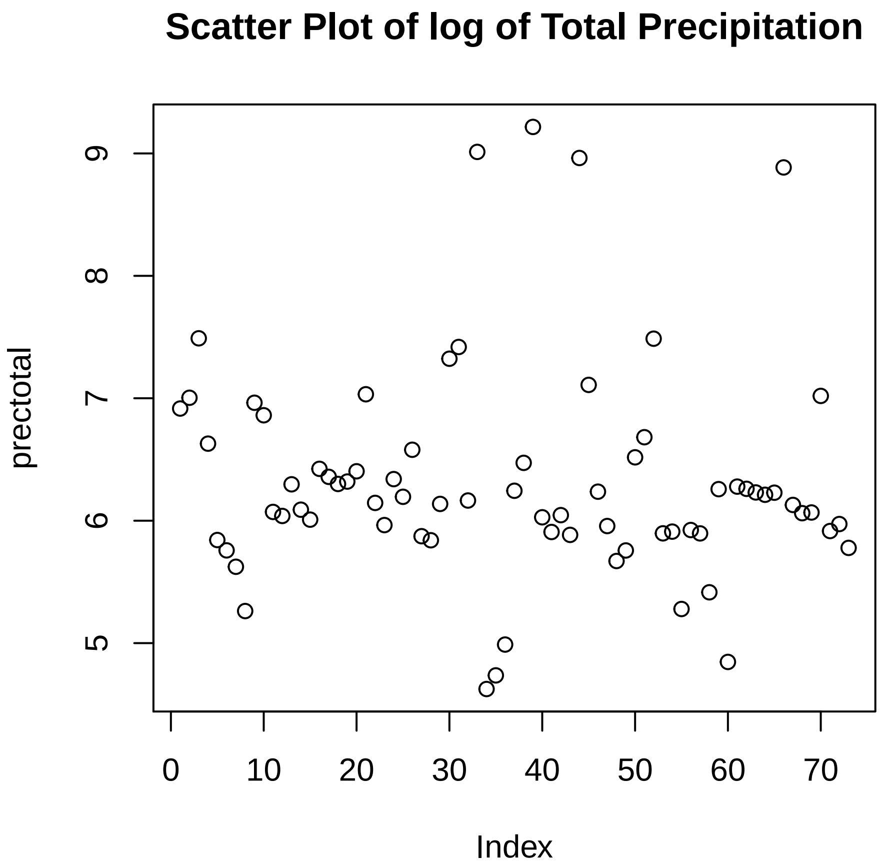

Figure 7: Scatter plot of Log of Total Precipitation.

In this section, we will fit a functional linear model where scalar response is the logarithm of total of precipitation, and functional covariate is temperature curves using Ramsay and Silverman's approach. In particular, we choose Bspline basis for the functional covariate, and without penalty. The generalized cross validation is used to obtain optimal number of basis functions. The optimal number of basis functions is 36. The R square of fitting predicted values by Ramsay and Silverman's approach vs. the true value of log of total precipitation is 0.6677309.

Moreover, we also use Principle Component Analysis to fit the linear model,the selection is performed by cross-validation to choose the best principle components. The optimal number of principle components used is 6, consisting of PC 1, PC 10, PC7, PC17, PC32, and PC25. The R square of fitting predicted values by PCA approach vs. the true value of log of total precipitation is 0.2219788.

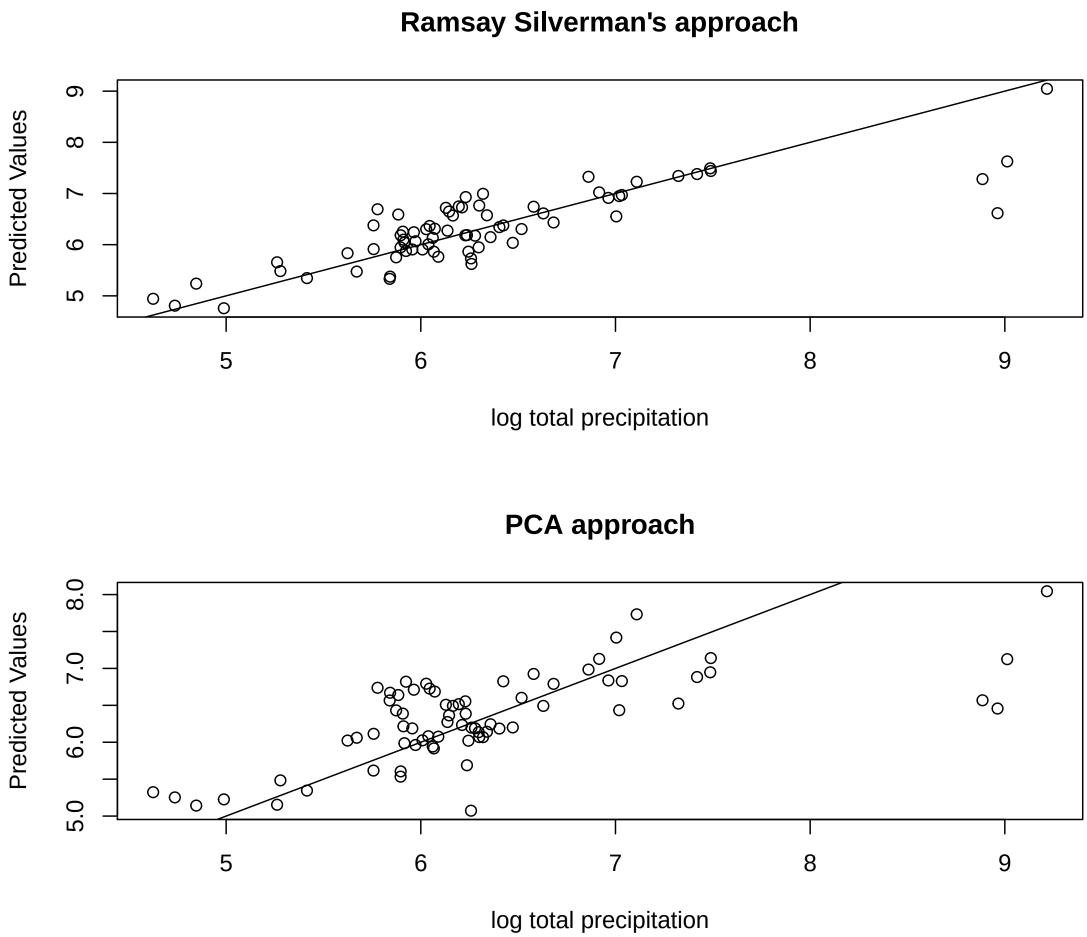

Figure 8: Fitted Values by Ramsay Silverman s Approach and PCA Approach.

The results of mean square errror and the Figure 8 supports the conclusion that the fitting of functional linear model using Ramsay and Silverman's approach has better prediction results comparing to the fitting using Principle Component Analysis approach.

5 Functional Linear Models with Functional Covariate as Temperature and Functional Response as Precipitation

In this section, we will fit a functional linear models with both covariate and response are functional data temperature and precipitation.

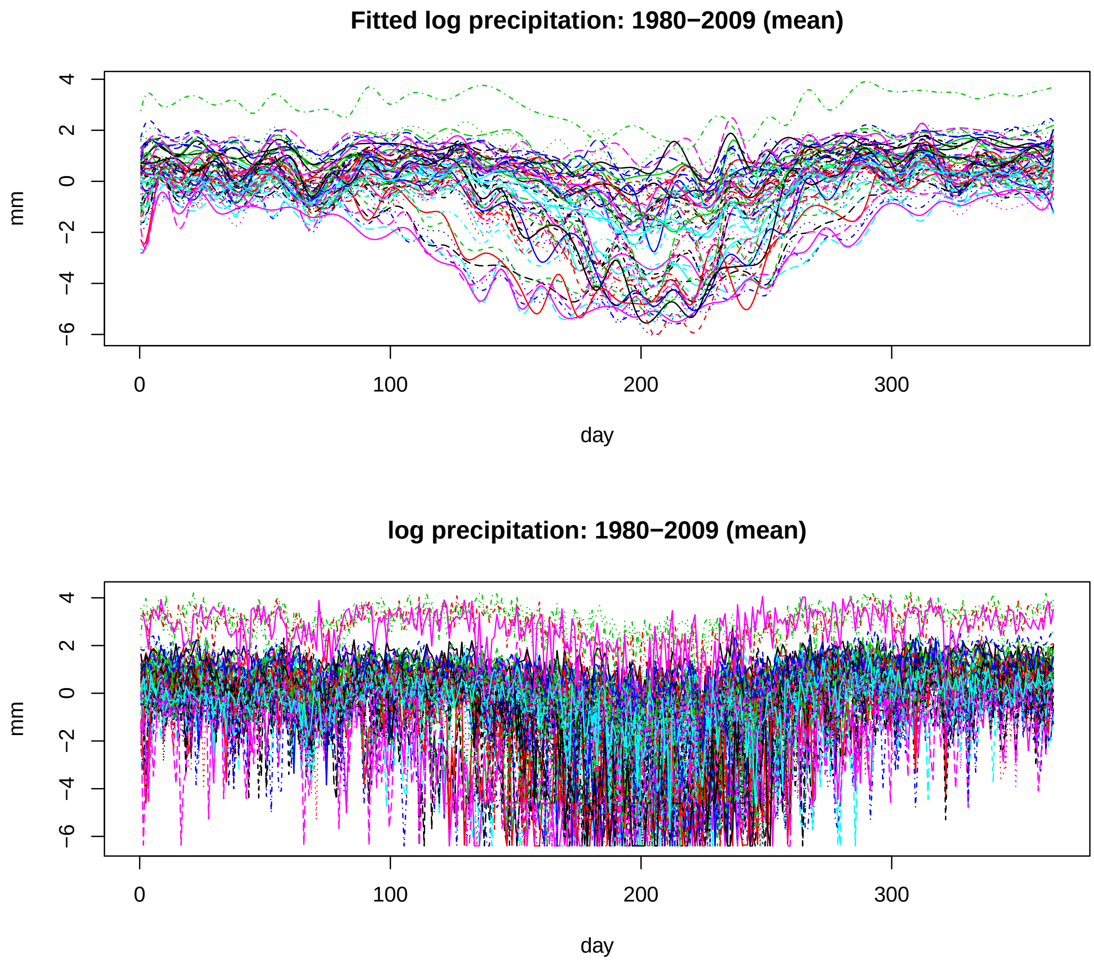

Figure 9: Fitted Curves using Functional Linear Model with Functional Response Precipitation vs. Functional data Temperature.

By looking at the Figure 9, we can see that the fitted predictioncurves capture very well the overall shape of the original data, the precipitation curves.

6 Conclusion

In this project, we apply some functional data analysis tools such as mean and covariance estimate, aw well as functional linear regression following Ramsay Silverman's approach and Principle Component Analysis. The outcome from the analysis suggest that Ramsay and Silverman's approach gives very promising results in estimating the mean and covariance matrix, and fitting functional linear model. The analysis shows there exist "negative" correlation between log precipitation and temperature. To the best of my knowlegde, there is no such analysis based on the `aemet` dataset, eventhough the analyses are very similar to what we have been done for the Canadian Weather dataset in the homework. The analyses carrying on in this project are still limitted in the estimating and prediction, but not doing any goodness of fit test or confidence intervals yet. One of the reason is that there hardly exists works done in this area to apply. Hopefully, more works will be available in this area in the future.

References

[1] Febrero-Bande, M., Oviedo de la Fuente, M. (2012). Statistical Computing in Functional Data Analysis: The R Package fda.usc. Journal of Statistical Software, 51(4), 1-28.

Appendices

The R code for analysis in this project is as below.

```
library(fda.usc)
library(reshape)
library(ggplot2)
##plot data
data(aemet)
par(mfrow=c(1,2))
plot(aemet$temp)
```

```r
plot(aemet$logprec)
dat2 <- cbind(melt(aemet$temp$data),
              logprec =melt(aemet$logprec$data)$value)
colnames(dat2)[3] <- "temperature"
ggplot(dat2, aes(x = temperature, y = logprec, group = as.factor(X1),
                 colour = as.factor(X1))) + geom_line() +
   guides(colour=FALSE)+ labs(title = "Logprec vs. Temperature")

#  -------------  set up fourier basis  ------------------------
#  Here it was decided that 65 basis functions captured enough of
#  the detail in the temperature data: about one basis function
#  per week.

#  The use of only 65 basis functions instead of 365
#  automatically generates some smoothing.

#  However, see below for smoothing with a saturated
#  basis (365 basis functions) where smoothing is defined by the
#  GCV criterion.

daybasis65 <- create.fourier.basis(rangeval=c(0, 365), nbasis=65)

#  -----------  set up the harmonic acceleration operator  ---

harmaccelLfd365 <- vec2Lfd(c(0,(2*pi/365)^2,0), c(0, 365))

#  ---------  create fd objects for temp. and prec. ----------

daytempfd <- smooth.basis(day.5, t(aemet$temp$data),
                          daybasis65,
                          fdnames=list("Day", "Station", "Deg C"))$fd
#  --  compute and plot mean and standard deviation of temperature -
(tempmeanfd  <- mean.fd(daytempfd))
(tempstdvfd  <- sd.fd(daytempfd))
op <- par(mfrow=c(2,2))
plot(tempmeanfd,               main="Mean")
plot(tempstdvfd, main="Standard Deviation", log="y")
```

```r
#  --  plot the temperature variance-covariance bivariate function  -
str(tempvarbifd <- var.fd(daytempfd))
str(tempvarmat  <- eval.bifd(weeks,weeks,tempvarbifd))
# dim(tempvarmat)= c(53, 53)

#op <- par(mfrow=c(1,2), pty="s")
#contour(tempvarmat, xlab="Days", ylab="Days")
contour(weeks, weeks, tempvarmat,
        xlab="Temperature by day",
        ylab="Temperature by day",
        main=paste("Variance function across stations"
                   ),
        cex.main=0.8, axes=FALSE)
axisIntervals(1, atTick1=seq(0, 365, length=5), atTick2=NA,
              atLabels=seq(1/8, 1, 1/4)*365,
              labels=paste("Q", 1:4) )
axisIntervals(2, atTick1=seq(0, 365, length=5), atTick2=NA,
              atLabels=seq(1/8, 1, 1/4)*365,
              labels=paste("Q", 1:4) )
#persp(tempvarmat,xlab="Days", ylab="Days", zlab="Covariance")
persp(weeks, weeks, tempvarmat,
      xlab="Days", ylab="Days", zlab="Covariance",
      main = "Covariance surface")
mtext("Temperature Covariance", line=-4, outer=TRUE)
par(op)
#################################################################

#  -------------------------------------------------------------------
#                PCA of temperatures with varimax rotation
#  -------------------------------------------------------------------

harmfdPar     <- fdPar(daybasis65, harmaccelLfd365, 1e5)
daytemppcaobj <- pca.fd(daytempfd, nharm=4, harmfdPar)
dimnames(daytemppcaobj$scores)[[2]] <- paste("PCA", 1:4, sep=".")
# plot principle components
op <- par(mfrow=c(1,2), pty="m")

#  plot harmonics/eigenfunctions
```

```
plot(daytemppcaobj$harmonics, main = "Eigenfunctions",
     xlab = "day")
#  plot log eigenvalues
daytempeigvals <- daytemppcaobj[[2]]
plot(1:20, log10(daytempeigvals[1:20]), type="b",
     xlab="Eigenvalue Number", ylab="Log 10 Eigenvalue",
     main = "Eigenvalues")
abline(lsfit(5:20, log10(daytempeigvals[5:20])), lty=2)

par(op)

# do the same thing for precipitation data
dayprecfd <- smooth.basis(day.5, t(aemet$logprec$data),
                          daybasis65,
                          fdnames=list("Day", "Station", "Logprec"))$fd
#  -- compute and plot mean and standard deviation of preceratur e -
precmeanfd  <- mean.fd(dayprecfd)
precstdvfd  <- sd.fd(dayprecfd)
op <- par(mfrow=c(2,2))
plot(precmeanfd, main="Mean")
plot(precstdvfd, main="Standard Deviation", log="y")
#  - plot the temperature variance-covariance bivariate function  -
precvarbifd <- var.fd(dayprecfd)
precvarmat  <- eval.bifd(weeks,weeks,precvarbifd)
contour(weeks, weeks, precvarmat,
        xlab="Precipitation by day",
        ylab="Precipitation by day",
        main=paste("Variance function across stations"
                   ),
        cex.main=0.8, axes=FALSE)
axisIntervals(1, atTick1=seq(0, 365, length=5), atTick2=NA,
              atLabels=seq(1/8, 1, 1/4)*365,
              labels=paste("Q", 1:4) )
axisIntervals(2, atTick1=seq(0, 365, length=5), atTick2=NA,
              atLabels=seq(1/8, 1, 1/4)*365,
              labels=paste("Q", 1:4) )
persp(weeks, weeks, precvarmat,
```

```
      xlab="Days", ylab="Days",
      zlab="Covariance",
      main = "Surface Covariance")
mtext("Precipitation Covariance", line=-4, outer=TRUE)
par(op)

#  ----------------------------------------------------------------
#              PCA of precipitation with varimax rotation
#  ----------------------------------------------------------------

harmfdPar     <- fdPar(daybasis65, harmaccelLfd365, 1e5)

dayprecpcaobj <- pca.fd(dayprecfd, nharm=4, harmfdPar)
dimnames(dayprecpcaobj$scores)[[2]] <- paste("PCA", 1:4, sep=".")
# plot principle components
op <- par(mfrow=c(1,2), pty="m")
#  plot harmonics/eigenfunctions
plot(dayprecpcaobj$harmonics, xlab = "day",
     main = "Eigenfunctions")
#  plot log eigenvalues
daypreceigvals <- dayprecpcaobj[[2]]
plot(1:20, log10(daypreceigvals[1:20]), type="b",
     xlab="Eigenvalue Number", ylab="Log 10 Eigenvalue",
     main = "eigenvalues")
abline(lsfit(5:20, log10(daypreceigvals[5:20])), lty=2)
par(op)
# plot of log of total precipitation
prectotal <- log(apply(exp(aemet$logprec$data), 1, sum))
plot(prectotal, main = "Scatter Plot of log of Total Precipitation")
# functional linear model with scalar response
fregre.cv.out <- fregre.basis.cv(aemet$temp, prectotal, lambda= TRUE)
fregre.cv.out$basis.x.opt
fregre.pc.cv.out <- fregre.pc.cv(aemet$temp, prectotal,  lambda = TRUE)
fregre.pc.cv.out$fregre.pc$l
plot(y = fregre.cv.out$fitt, x = prectotal,
     xlab ="log total precipitation",
     main = "Ramsay Silverman's approach",
     ylab = "Predicted Values")
```

```
abline(a=0,b=1)
plot(y = fregre.pc.cv.out$fregre.pc$fitt, x = prectotal,
     xlab ="log total precipitation",
     main = "PCA approach",
     ylab = "Predicted Values")
abline(a=0,b=1)
summary(lm(fregre.cv.out$fitt~ prectotal))$r.squared
summary(lm(fregre.pc.cv.out$fregre.pc$fitt~ prectotal))$r.squared
# functional linear model with functional response
out.fregre.fr <- fregre.basis.fr(aemet$temp, aemet$logprec)
plot(out.fregre.fr$fitted, main = "Fitted
     log precipitation: 1980-2009 (mean)")
plot(aemet$logprec)
```