

Studforsk Project  
Small Study on Penalized Complexity Priors for stationary auto  
regressive (AR) for order 1

Himanshu Srivastav

October 27, 2016

# Contents

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Objective</b>	<b>1</b>
<b>3</b>	<b>Introduction about AR(1) process</b>	<b>1</b>
<b>4</b>	<b>Penalized Complexity Priors</b>	<b>1</b>
<b>5</b>	<b>Construction of PC priors for lag-one correlation in an AR(1) model</b>	<b>4</b>
<b>6</b>	<b>Fitting of AR(1) data points using PC priors</b>	<b>8</b>
<b>7</b>	<b>Comparison of PC Priors with Reference and Jeffery's Priors</b>	<b>20</b>
<b>8</b>	<b>Conclusions/Discussions</b>	<b>26</b>
<b>9</b>	<b>Acknowledgement</b>	<b>26</b>

# 1 Abstract

In Bayesian statistical inference of time series models, we assign prior distribution for all hyper parameters. These prior distribution about the hyper parameters represents our prior beliefs/understanding about the hyper parameters space. In general, it is very hard to express exact prior information about the hyper parameters. Or requires an expert knowledge to mention the concrete probabilistic information about hyper parameters. Penalized Complexity priors (PC priors) are weakly informative priors, where we have some useful information about the parameters. In this report, we will apply PC priors on auto regressive process of order 1, which are a basic model for AR(p). PC priors is based on Occam's razor principle. They exhibit important properties like invariance under reparameterization, robustness properties.

## 2 Objective

In this report, we are going to explain: 1. Fundamentals for Construction of PC priors 2. How to construct PC Priors for auto correlation at lag one of AR(1) process. 3. Fitting of Simulated Data using PC Priors 4. Behavior of Jeffrey's and Reference prior distribution w.r.t the measure of complexity for AR(1) process

## 3 Introduction about AR(1) process

AR(1) process is generally defined by following:

$$x_t = \phi x_{t-1} + w_t, \quad (1)$$

where  $w_t \sim \mathcal{N}(0, \kappa^{-1})$ , for  $t = 2, 3, \dots, n$ .  $x_1$  is assumed to follow mean 0 and marginal precision  $\tau = \kappa(1 - \phi^2)$ . In AR(1) process dependency is governed by the factor  $\phi$ , for stationary AR (1) process, we have to limit  $|\phi| < 1$ . We will formulate prior distribution for parameter  $\phi$  using PC priors. We will consider two choices for the base model depending on our application we may choose any one them. We can choose base model to be independent wrt time ie no dependency in time that corresponds to  $\phi = 0$ . And we can also choose base model to be no change wrt to time that corresponds to the case when  $\phi = 1$ .

## 4 Penalized Complexity Priors

Penalized complexity priors [1] assigns prior distributions for hyper parameters, based on the complexity of the model. As the name suggests, PC priors penalizes priors for having more complex model. Priors for base model(simpler model) are more probable then the priors for the complex models. This is the basic idea for construction of Penalized Complexity priors. We formulate PC priors using following principles:

### Principle 1: Occam's Razor:

Principle of parsimony explains that simpler model formulations should be preferred until there is enough support for a more complex model. For our frame of working, the simpler model is base model, so we will penalize priors for deviating from the base model. This hints that as the complexity of the flexible model increases priors for the flexible model will be less probable when compared to the priors for the the base model, which imply that prior densities of hyper parameters should decay as the complexity (measure of distance between two model) increases. But there arises the question how to measure the distance between two models or what are the measure of complexity in our frame of working. We introduce next principle for measuring the complexity between to models.

### Principle 2: Measure of Complexity:

We use Kullback -Liebler divergence (KLD) to measure the increased complexity. Between to densities  $f$  and  $g$ , KLD is defined by

$$KLD(f||g) = \int f(x) \log \frac{f(x)}{g(x)} dx$$

KLD is a measure of the information lost when the base model  $g$  is used to approximate model  $f$ . Note, that this is an asymmetric function, which means measure of complexity is non symmetric, hence the distant considered by KLD is not metric. Since, this measure of complexity is generated through integration of densities, it doesn't match with notion of distance dimensionally. Hence, we will use the unidirectional measure of distance  $d(f||g) = \sqrt{(2KLD(f||g))}$ .  $d$  is considered to be measure of complexity model  $f$  when compared to model  $g$ . The factor of "2" is chosen for the sake of convenience. Now, we have established a suitable measure of distance between base model and flexible model, in the form of KLD measure, and we know that priors are to be penalized wrt to this measure of distance from the base model. But, we are not still prepared to construct the PC prior, as we don't know how should we penalize prior distribution quantitatively, which lead us to the 3rd principle.

### Principle 3: Constant rate penalization:

Prior for the distance measure  $d$ , it is natural to assume that the mode should be located at the base model while the density decays as the distance from the base model increases, so that the prior densities satisfy following

$$\frac{\pi_d(d + \delta)}{\pi_d(d)} = r^\delta, d, \delta \geq 0$$

for some constant  $0 < r < 1$ . The idea behind constant rate penalization is that relative change in prior densities doesn't depend on  $d$  when we increase the complexity of flexible model from  $d$  to  $d\delta$ . Since we know that  $d = 0$  represent our base model, this idea of constant rate penalization assumption implies an exponential prior on the distance scale. So we have following distributions for prior distribution

$$\pi_d(d) = \lambda \exp(-\lambda d)$$

for  $r = \exp(-\lambda)$ . We should note here that we measure complexity using KLD, we will get distance between base model and flexible model  $d$  as a function of  $\xi$ . lets say we denote  $d$  as a function of  $\xi$  as  $d(\xi)$ . Now, we know from constant rate penalization that  $d(\xi) \sim \exp(\lambda)$ . So we need to transformation of random variable to find out the prior distributions in the following way

$$\pi(\xi) = \pi(d(\xi)) \left| \frac{\partial d(\xi)}{\partial \xi} \right|$$

We have constructed PC prior for our hyper parameter of interest  $\xi$ , PC priors density is also a function of  $\lambda$ , which is unknown as of now. We have not specify anything about the parameter  $\lambda$  till now. So we introduce our next principle for interpretation of  $\lambda$ .

### Principle 4: User defined scaling:

We started our discussion by saying that PC priors are the weakly informative priors, however till now we have not introduced any information about the hyper parameters, construction till now is quite generic and may be applied to a class of hyper parameters, however in the real life situation we may need to apply it to particular problem. We may have broad idea about the sensible upper bound  $U$ -sensible user defined upper bound and  $\alpha$  the tail event we put about this sensible upper bound. With the use of  $U, \alpha$  we also give a significance to the unknown parameter  $\lambda$ , we use this prior knowledge in the following manner  $Prob(Q(\xi) > U) = \alpha$ , where  $Q(\xi)$  is an interpret-able transformation of the flexibility parameter.

We should note that the idea of PC prior is very useful, when we have some vague idea of the hyper parameters in the following form, it is unlikely that parameter is larger than some number. Now, we are prepared for construction of a PC prior, we will construct PC prior for the precision parameter for a multivariate normal distribution as an example before starting it lag one correlation for AR(1) process.

### Example 1: Construction of PC priors for precision in multivariate Normal distribution:

Let  $\mathcal{N}_0^p(\mu_0, \Sigma_0)$  denote our base model which follows a multivariate normal distribution with dimension  $p$ . And our flexible model is of the form  $\mathcal{N}_1^p(\mu_1, \Sigma_1)$ , using KLD to calculate the measure of distance between two models,

$$KLD(\mathcal{N}_1^p || \mathcal{N}_0^p) = \frac{1}{2} \{tr(\Sigma_0^{-1}\Sigma_1) + (\mu_0 - \mu_1)^T \Sigma_0^{-1}(\mu_0 - \mu_1) - p - \log \frac{|\Sigma_1|}{|\Sigma_0|}\}$$

We are formulating PC priors for precision parameters, so simplest assumption for the base model will be a model with no random effect and for flexible model, we are adding random effect to our base model. i.e.  $\Sigma_0 = 0$ , which is not useful directly for calculating the distance between two models of interest so, we will assume  $\Sigma_0 = \frac{R}{\tau_0}$  and  $\Sigma_1 = \frac{R}{\tau}$  then we will take limit

$$\lim \tau_0 \rightarrow \infty$$

R is assumed to be full rank fixed matrix. Since, both models differs wrt to precision parameter so we assume them to have same means. Using all these we estimated following

$$KLD = \frac{p}{2} \frac{\tau_0}{\tau} \left\{ 1 + \frac{\tau}{\tau_0} \log\left(\frac{\tau}{\tau_0}\right) - \frac{\tau}{\tau_0} \right\}$$

Now, as  $\tau_0 \gg \tau$  ie equivalent to that when  $\tau_0$  goes to  $\infty$  the  $\frac{\tau}{\tau_0}, \frac{\tau}{\tau_0} \log\left(\frac{\tau}{\tau_0}\right)$  will go to 0, which gives us

$$d(\tau) = \sqrt{\frac{p\tau_0}{\tau}}$$

Now, we have distance between two models in terms of hyper parameter, we can apply principle 3 and principle 4 to get the exact prior distribution for precision parameter. So we know that  $d(\tau)$  as a function of  $\tau$  and also knows that it follows exponential distribution with rate  $\lambda$  so by transforming the variables, we get the prior distribution for our parameter of interest in the following way:

$$\left| \frac{\partial d(\tau)}{\partial \tau} \right| = \frac{1}{2} \sqrt{\frac{p\tau_0}{\tau^3}}$$

$$\pi(d(\tau)) = \lambda \exp -\lambda d(\tau)$$

which gives us following prior distribution for our parameter of interest:

$$\pi(\tau) = \frac{\theta}{2} \tau^{-\frac{3}{2}} \exp \frac{-\theta}{\sqrt{\tau}}$$

where  $\theta = (-\lambda\sqrt{p\tau_0})$ . This prior distribution is type -2 Gumbel distribution. To find the significance of  $\theta$ , we need to apply principle 4. i.e. probability statement

$$Prob\left(\frac{1}{\sqrt{\tau}} > U\right) = \alpha$$

which gives us that

$$\theta = -\frac{\log \alpha}{U}$$

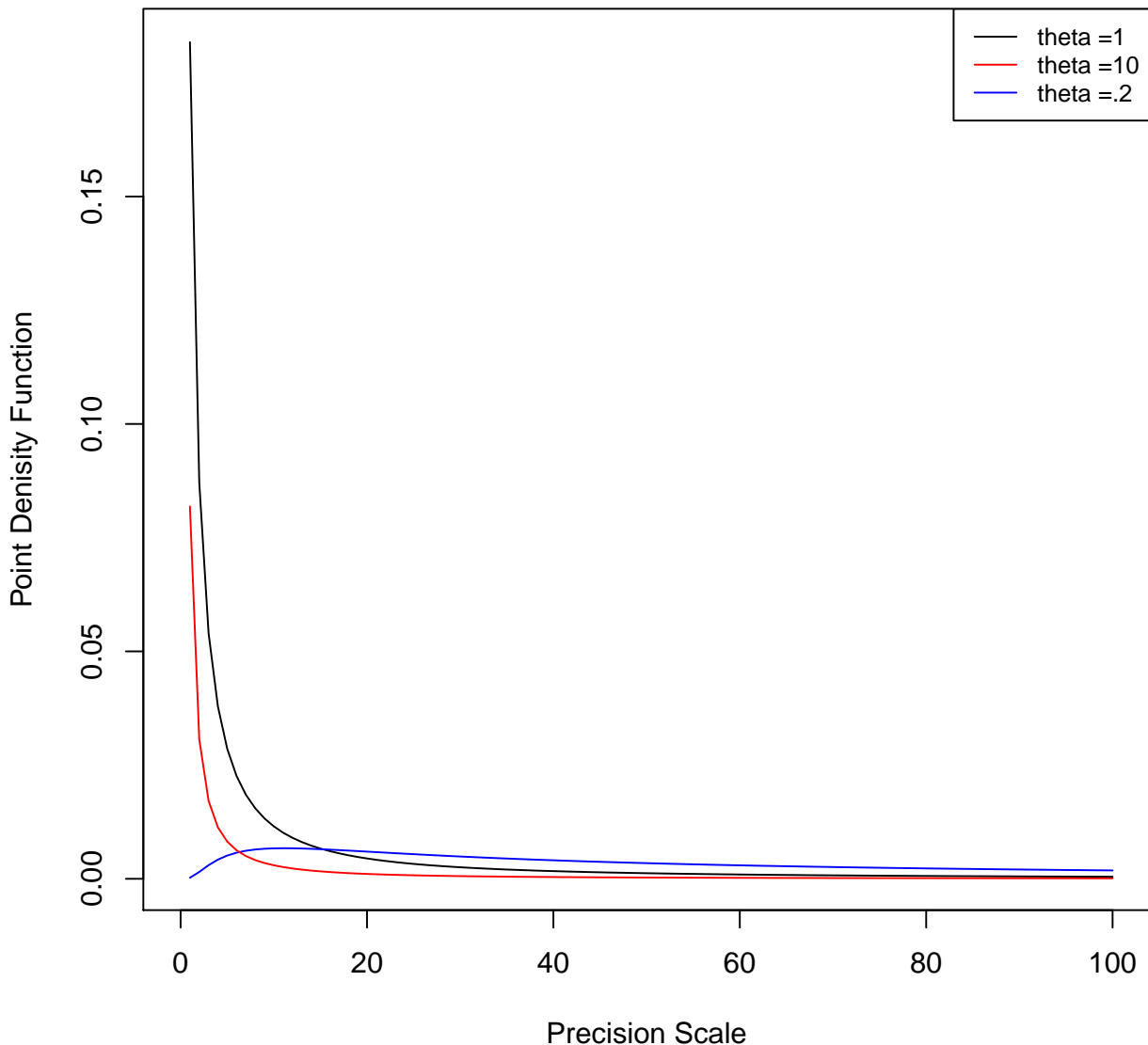
Plots of PC priors for precision are as follows, we have set three different values of  $\theta$  ie 1, 10, .2

```
curve((1/2)*x^(-3/2)*exp(-1/sqrt(x)), from = 0, to = 100,
      xlab = "Precision Scale",
      ylab = "Point Denisity Function")
curve((10/2)*x^(-3/2)*exp(-10/sqrt(x)), from = 0, to = 100, add = TRUE, col = "blue")
curve((.2/2)*x^(-3/2)*exp(-.2/sqrt(x)), from = 0, to = 100, add = TRUE, col = "red")

title(main = "PC Priors for precision parameter")

legend("topright", legend=c("theta =1", "theta =10", "theta =.2"),
      col=c("black", "red", "blue"), lty=c(1,1,1), cex=0.8)
```

## PC Priors for precision parameter



We observe that for various values of  $\theta$ , PC priors behaves both like the informative priors and uninformative priors as well. In the Figure above we see that when  $\theta = 1$  then, PC priors shows shrinkage and becomes informative where as in the case when  $\theta$  was set to .2 they behaves like flat priors.

## 5 Construction of PC priors for lag-one correlation in an AR(1) model

AR(1) process is generally defined by following:

$$x_t = \phi x_{t-1} + w_t, \quad (2)$$

where  $w_t \sim \mathcal{N}(0, \kappa^{-1})$ , for  $t = 2, 3, \dots, n$ .  $x_1$  is assumed to follow mean 0 and marginal precision  $\tau = \kappa(1 - \phi^2)$ . In AR(1) process dependency is governed by the factor  $\phi$ , for stationary AR (1) process, we have to limit  $|\phi| < 1$ . We have two choices for the base model depending on our application we may choose any one them. We can choose base model to be independent wrt time ie no dependency in time that corresponds to  $\phi = 0$ . And we can also choose base model to be no change wrt to time that corresponds to the case when  $\phi = 1$ . Construction of PC priors will follow the same principles, however both approaches will lead to different prior distributions. For simplification of our calculation, we assume that we know precision for the noise.

## Construction of PC priors for Base Model: No dependency in time

let say  $p$  s dimension of data, since we are considering the first case for  $\phi = 0$  for our base model and  $\phi$  for our flexible model. Since for calculating the KLD and  $d(\phi)$  we need  $\mu_0$  and  $\mu_1$  and  $\Sigma_0$  and  $\Sigma_1$ .  $\mu_0 = \mu_1 = 0$  and  $\Sigma_0$  is an identity matrix of order  $p$ , and  $\Sigma_1$  is such that  $(\Sigma_{ij}) = \frac{1}{\tau}\phi^{|i-j|}$ ,

$$\Sigma_1 = \frac{1}{\tau} \begin{bmatrix} 1 & \phi & \phi^2 & \dots & \phi^{p-1} \\ \phi & 1 & \phi^2 & \dots & \phi^{p-2} \\ \phi^2 & \phi & 1 & \dots & \phi^{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi^{p-1} & \phi^{p-2} & \phi^{p-3} & \dots & 1 \end{bmatrix}$$

using the following equation we can found the KLD between base model and the flexible model:

$$KLD(\mathcal{N}_1^p || \mathcal{N}_0^p) = \frac{1}{2} \{tr(\Sigma_0^{-1}\Sigma_1) + (\mu_0 - \mu_1)^T \Sigma_0^{-1}(\mu_0 - \mu_1) - p - \log \frac{|\Sigma_1|}{|\Sigma_0|}\}$$

It comes to be:

$$KLD(\mathcal{N}_1^p || \mathcal{N}_0^p) = \frac{1}{2}(1-p) \log(1-\phi^2)$$

and  $d(\phi) = \sqrt{(1-p) \log(1-\phi^2)}$ . Since, we have now the distance as a function of  $\phi$  and by using the principle 3 we can find the prior distribution for the  $\phi$  in the following way:

$$\left| \frac{\partial d(\phi)}{\partial \phi} \right| = \frac{1}{2} \sqrt{\frac{1-p}{\log(1-\phi^2)}} 2 \frac{|\phi|}{(1-\phi^2)}$$

$$\pi(d(\phi)) = \lambda \exp(-\lambda d(\phi))$$

Using them we formulated following prior distribution for  $\phi$  [2]:

$$\pi(\phi) = \theta \exp(-\theta \sqrt{-\log(1-\phi^2)}) \frac{\phi}{(1-\phi^2) \sqrt{-\log(1-\phi^2)}}$$

where  $\theta = \lambda \sqrt{p-1}$ . To find our significance of  $\theta$ , we have to use principle 4, so we define the probability statement  $Prob(|\phi| > U) = \alpha$ , which gives us that

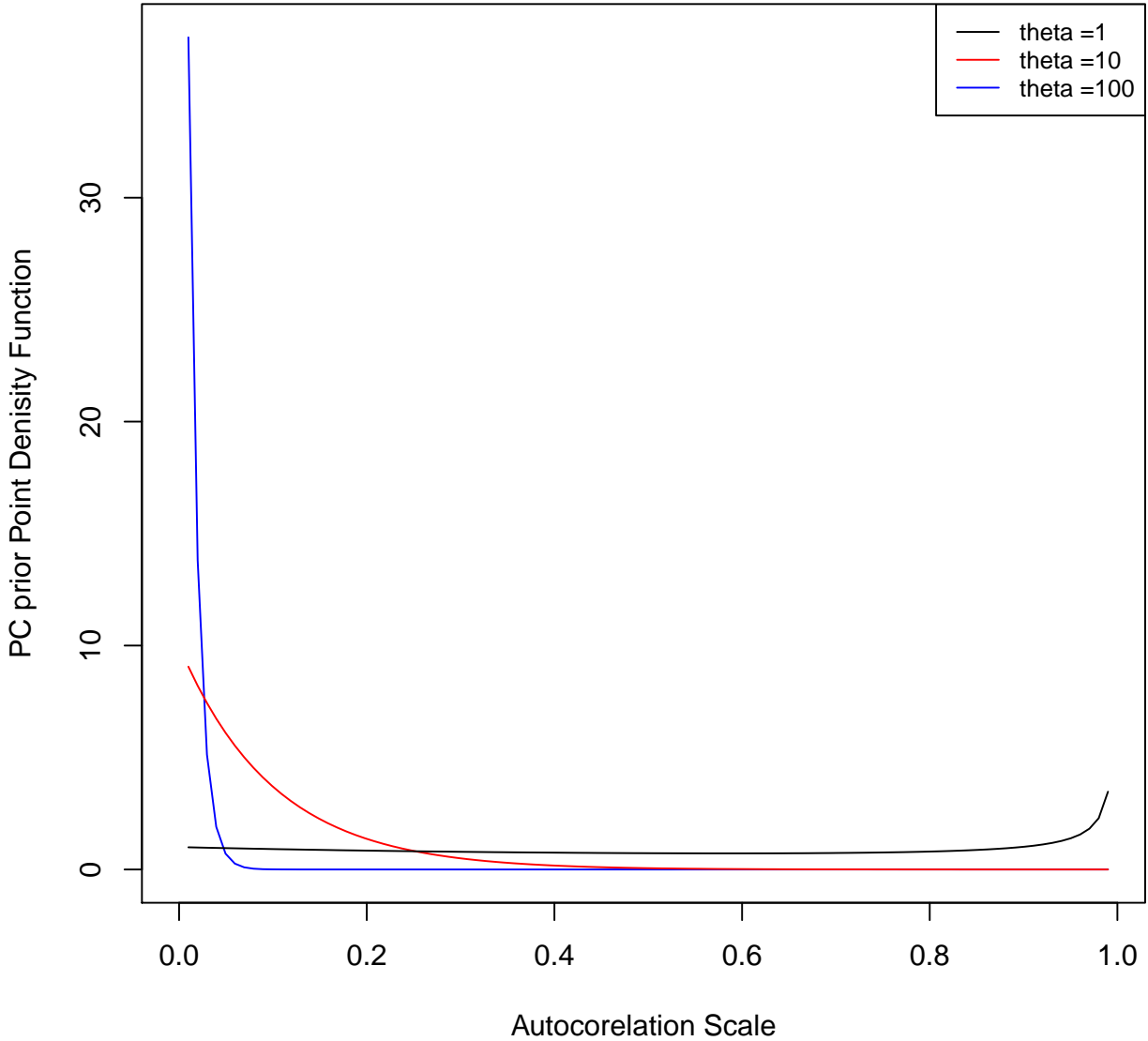
$$\theta = \frac{-\log \alpha}{\sqrt{-\log(1-U^2)}}$$

Plots of PC priors for correlation at lag one are as follows, we have set three different values of  $\theta$  ie 100, 10, 1

```
curve((100)*exp(-100*sqrt(-log(1-x^2)))*x/(1-x^2)/sqrt(-log(1-x^2)),
      from = 0, to = .99,
      col = "blue", xlab = "Autocorelation Scale",
      ylab = "PC prior Point Denisity Function")
curve((10)*exp(-10*sqrt(-log(1-x^2)))*x/(1-x^2)/sqrt(-log(1-x^2)),
      from = 0, to = .99, col = "Red", add = TRUE)
curve((1)*exp(-1*sqrt(-log(1-x^2)))*x/(1-x^2)/sqrt(-log(1-x^2)),
      from = 0, to = .99, col = "Black", add = TRUE)

legend("topright", legend=c("theta =1", "theta =10", "theta =100"),
      col=c("black", "red", "blue"), lty=c(1,1,1), cex=0.8)
title(main = "PC Prior for autocorrelation when distance calculated
          considering phi =0 the base model")
```

## PC Prior for autocorrelation when distance calculated considering phi =0 the base model



We observe that for various values of  $\theta$ , PC priors behaves both like the informative priors and uninformative priors. In the Figure above we see that when  $\theta = 100$  then, PC priors shows shrinkage and becomes informative about where as in the case when it was set to 1 they behaves like flat priors.

### Construction of PC priors for Base Model: No change in time

An alternative approach is for base model is that base model does not change with time ie  $\phi = 1$ . This represents the limiting case of random walk, which is non stationary and singular process. so we will assume  $\phi_0$  for the base model and  $\phi$  for the flexible model, such that  $\phi_0 > \phi$ . And will discuss the limiting case when limit

$$\lim \phi_0 \rightarrow 1.$$

So  $\Sigma_0$  and  $\Sigma_1$  for our base and flexible models looks like following

$$\Sigma_0 = \frac{1}{\tau} \begin{bmatrix} 1 & \phi_0 & \phi_0^2 & \dots & \phi_0^{p-1} \\ \phi_0 & 1 & \phi_0^2 & \dots & \phi_0^{p-2} \\ \phi_0^2 & \phi_0 & 1 & \dots & \phi_0^{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_0^{p-1} & \phi_0^{p-2} & \phi_0^{p-3} & \dots & 1 \end{bmatrix} \quad \Sigma_1 = \frac{1}{\tau} \begin{bmatrix} 1 & \phi & \phi^2 & \dots & \phi^{p-1} \\ \phi & 1 & \phi^2 & \dots & \phi^{p-2} \\ \phi^2 & \phi & 1 & \dots & \phi^{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi^{p-1} & \phi^{p-2} & \phi^{p-3} & \dots & 1 \end{bmatrix}$$



This gives

$$KLD(\mathcal{N}_1^p || \mathcal{N}_0^p) = \frac{1}{2} \left[ \frac{1}{1 - \phi_0^2} \{p - 2(p-1)\phi_0\phi + (p-2)\phi_0^2\} - p - (p-1) \log \frac{(1 - \phi^2)}{(1 - \phi_0^2)} \right]$$

While considering the limiting case  $\lim \phi_0 \rightarrow 1$ , we have

$$d(\phi) = \sqrt{2KLD} = \sqrt{\frac{2(p-1)(1-\phi)}{1-\phi_0^2}} = c\sqrt{1-\phi},$$

where  $|\phi| < 1$  and  $c$  is independent from  $\phi$ . It is also to be noted that  $d(\phi) \leq c\sqrt{2}$ . Since, our  $d(\phi)$  now has a range so we have to use truncated exponential distribution. So in this case, we can construct prior for  $\phi$  in the following manner [2]:

$$\begin{aligned} d(\phi) &= c\sqrt{1-\phi} \\ \pi(\phi) &= \frac{\lambda \exp(-\lambda d(\phi))}{1 - \exp(-\sqrt{2}\lambda c)} \left| \frac{\partial d(\phi)}{\partial \phi} \right| = \frac{c}{2\sqrt{1-\phi}} \\ \pi(\phi) &= \frac{1}{2} \frac{\theta \exp(-\theta\sqrt{1-\phi})}{(1 - \exp(-\sqrt{2}\theta))\sqrt{1-\phi}} \end{aligned}$$

where  $\theta = \lambda c$ .

However for the significance of parameter  $\theta$  we have to use principle 4 i.e. probability statement

$$Prob(|\phi| > U) = \alpha,$$

which gives us following relationship, which can be solved for  $\theta$  :

$$\alpha = \frac{1 - \exp(-\theta\sqrt{1-U})}{(1 - \exp(-\sqrt{2}\theta))}$$

We must note that lower limit for

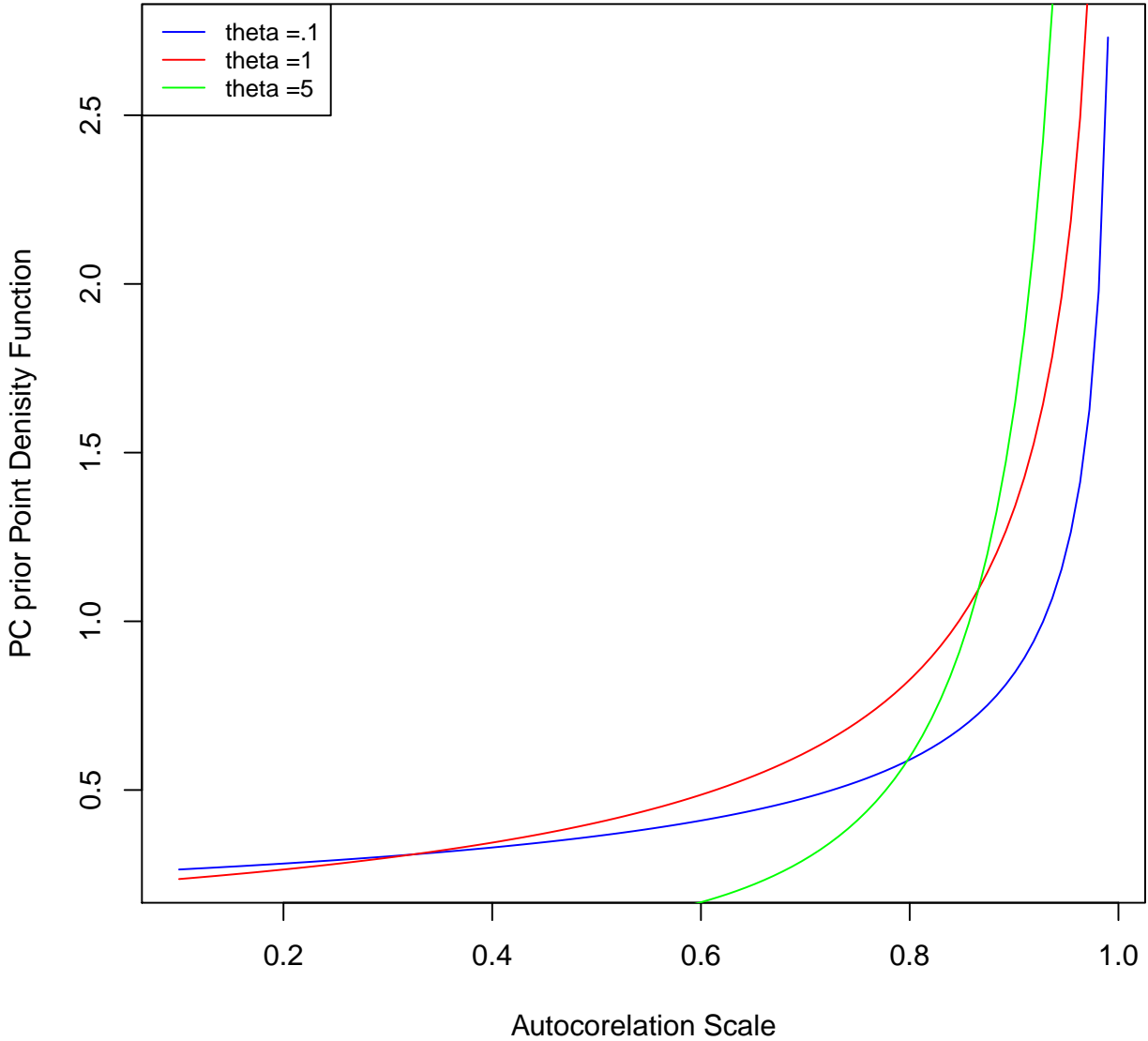
$$\alpha_{min} = \sqrt{\frac{1-U}{2}}$$

Plots of PC priors for correlation at lag one are as follows, we have set three different values of  $\theta$  ie 100, 10, 1

```
curve(.5*(.1)*exp(-.1*sqrt(1-x))/(1-exp(-2*.1))/sqrt(1-x),
      from = .1, to = .99,
      col = "blue",
      xlab = "Autocorelation Scale", ylab = "PC prior Point Denisity Function")
curve(.5*(1)*exp(-1*sqrt(1-x))/(1-exp(-2*1))/sqrt(1-x),,
      from = 0.1, to = .99, col = "Red", add = TRUE)
curve(.5*(5)*exp(-5*sqrt(1-x))/(1-exp(-2*5))/sqrt(1-x),
      from = 0.1, to = .99, col = "Green", add = TRUE)

legend("topleft", legend=c("theta =.1", "theta =1", "theta =5"),
      col=c("blue", "red", "Green"), lty=c(1,1,1), cex=0.8)
title(main = "PC Prior for autocorrelation when distance calculated
          considering phi =1 the base model")
```

## PC Prior for autocorrelation when distance calculated considering phi =1 the base model



We observe that for various values of  $\theta$ , PC priors behaves both like the informative priors and uninformative priors. In the Figure above we see that increasing values of  $\theta$  increases the amount of information about the data, which corresponds to more shrinkage towards the base model. However, by changing user defined scale, we may construct them as uninformative priors also.

## 6 Fitting of AR(1) data points using PC priors

Since, we have constructed the PC priors for lag one correlation for AR(1) process analytically. Now, we would like to simulate and see, how well does they fit to data. We have considered following cases:

### Case 1 : Effect of user defined scale when PC priors derived assuming base model to be time independent ( $\phi = 0$ ):

We have simulated a time series data of an AR(1) process, using the  $\phi = 0.4$ . Tried to fit it using various user defined scale to see the effect of user defined scales on priors, we have considered three user defined scales, please note that the probability statement in this case is as follows

$$Prob(|\phi| > U) = \alpha,$$

we have defined three scales for  $\alpha = (0.5, 0.05, 0.005)$ :

```

####Case 1 for different values of use defined
###scale when base model is considered with model rho =0

require(MASS)

## Loading required package: MASS

require(sos)

## Loading required package: sos
## Loading required package: brew
##
## Attaching package: 'sos'
##
## The following object is masked from 'package:utils':
##
##      ?

library(INLA)

## Warning: package 'INLA' was built under R version 3.2.3
## Loading required package: sp
## Warning: package 'sp' was built under R version 3.2.4
## Loading required package: Matrix
## Loading required package: splines
## This is INLA 0.0-1457953075, dated 2016-03-14 (11:43:05+0100).
## See www.r-inla.org/contact-us for how to get help.

library(spam)

## Warning: package 'spam' was built under R version 3.2.4
## Loading required package: grid
## Spam version 1.3-0 (2015-10-24) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
## The following object is masked from 'package:INLA':
##
##      Oral
##
## The following objects are masked from 'package:base':
##
##      backsolve, forwardsolve

library(MASS)
n = 50
rho = 0.4
s = 0.1

#simulate a AR(1) data for a given marginal precision and rho

simulated_ts = scale(as.numeric(arima.sim(n = n, list(ar=c(rho)))))+ rnorm(n, sd=s)

# define values for a suitable guess
u = .4

```

```

alpha = c(0.5, 0.05 , 0.005)

# define prior distribution for hyper parameters
# define data frame as a input to inla
data = list(y=simulated_ts, z=1:n)

#for (i in 1:3) {

formula1 = simulated_ts~ -1 + f(z,model="ar1",
                               hyper = list(rho = list(prior = "pc.cor0",
                                                       param = c(u, 0.5)),
                               prec = list(initial = log(1), fixed=TRUE)))

formula2 = simulated_ts~ -1 + f(z,model="ar1",
                               hyper = list(rho = list(prior = "pc.cor0",
                                                       param = c(u, 0.05)),
                               prec = list(initial = log(1), fixed=TRUE)))

formula3 = simulated_ts~ -1 + f(z,model="ar1",
                               hyper = list(rho = list(prior = "pc.cor0",
                                                       param = c(u, 0.005)),
                               prec = list(initial = log(1), fixed=TRUE)))

r1 = inla (formula1, data = data,
          control.family = list(hyper = list(
            prec = list(initial = log(1/s^2), fixed = TRUE))))

r2 = inla (formula2, data = data,
          control.family = list(hyper = list(
            prec = list(initial = log(1/s^2), fixed = TRUE))))

r3 = inla (formula3, data = data,
          control.family = list(hyper = list(
            prec = list(initial = log(1/s^2), fixed = TRUE))))
r1=inla.hyperpar(r1)
r2=inla.hyperpar(r2)
r3=inla.hyperpar(r3)

```

We can see from Figure 1 that PC Priors behaves like a flat prior, as we have used weakly informative user defined scale. We have allowed 50% of probability mass for  $\phi$  from the maximum limit of 0.4 for  $\phi$ . We have got following estimates in this case

```

r1$summary.hyperpar

##              mean          sd 0.025quant 0.5quant 0.975quant      mode
## Rho for z 0.4463463 0.1081239 0.2064684 0.45689 0.6275375 0.4792881

```

When we change the user defined scale to the a more informative condition ie we have set  $\alpha = 0.05$ , we have got plots in the Figure 2. We see that PC Priors shows excellent shrinkage towards base model. Since, the prior are are more informative then flat priors in this case, we can see that estimates for  $\phi$  are closer towards base model. We have got following estimates in this case

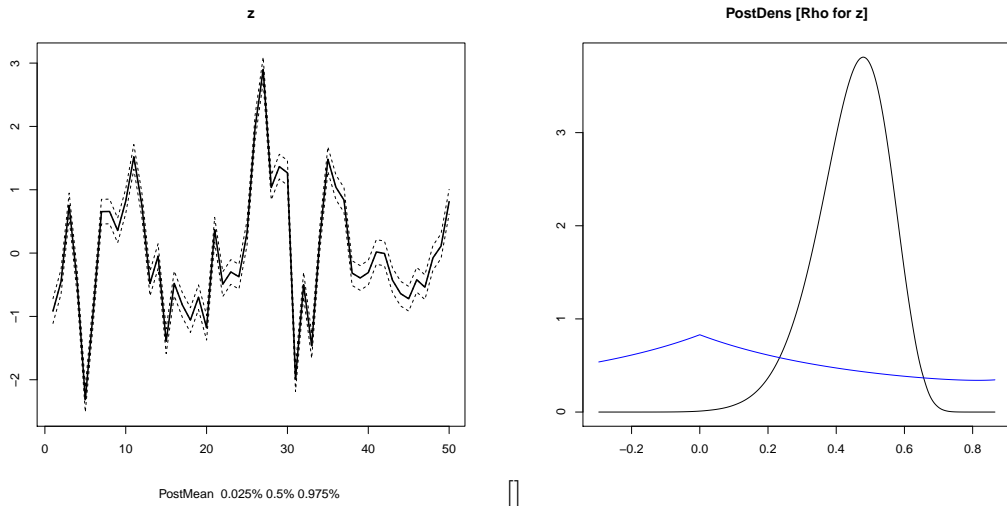


Figure 1: Plot of PC Priors(in Blue) when user defined scales are  $U = 0.4$  and  $\alpha = 0.5$

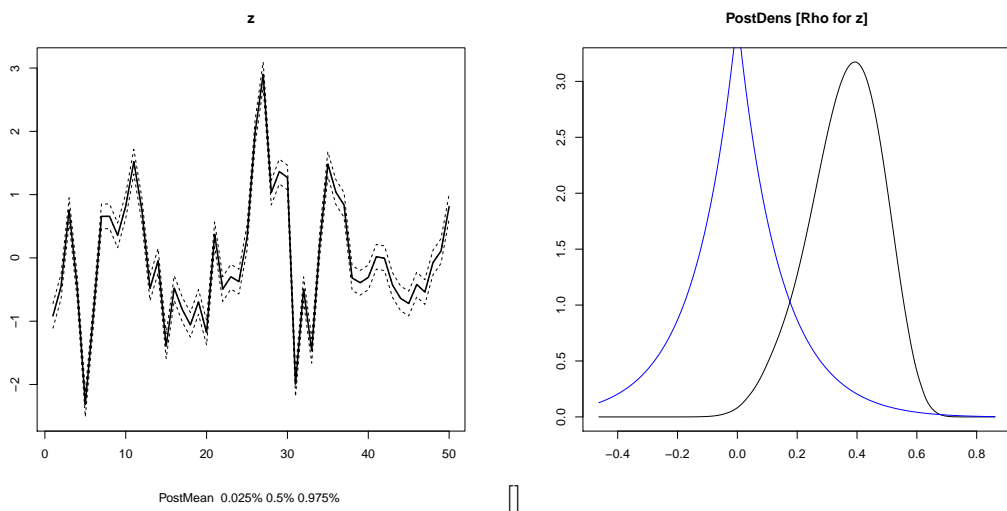


Figure 2: Plot of PC Priors(in Blue) when user defined scales are  $U = 0.4$  and  $\alpha = 0.05$

```
r2$summary.hyperpar
```

```
##          mean          sd 0.025quant  0.5quant 0.975quant    mode
## Rho for z 0.36053 0.1244881 0.09416577 0.3700227 0.5766252 0.3929747
```

We have changed the user defined scale to a stronger condition in this case, we are allowing only 0.5 % probability mass of  $\phi$  to be away from the maximum limit of 0.4. We expect stronger shrinkage for priors and estimates for  $\phi$  will be more close to the base models. We have got plots in the form of Figure 3 in this case. We observe that PC Priors are exhibiting the strong shrinkage towards base models and estimates for  $\phi$  are more closer towards base model. We have got following estimates in this case

```
r3$summary.hyperpar
```

```
##          mean          sd 0.025quant  0.5quant 0.975quant    mode
## Rho for z 0.2583546 0.1319289 0.01679603 0.2613675 0.5073127 0.2807103
```

We observed that PC Priors can be both highly informative or non informative based on the idea we have about our data. Our estimation for parameter of interest depends on prior distribution for the parameter of interest. We should note that our estimation will be more towards the base model if we have more shrinkage for our priors.

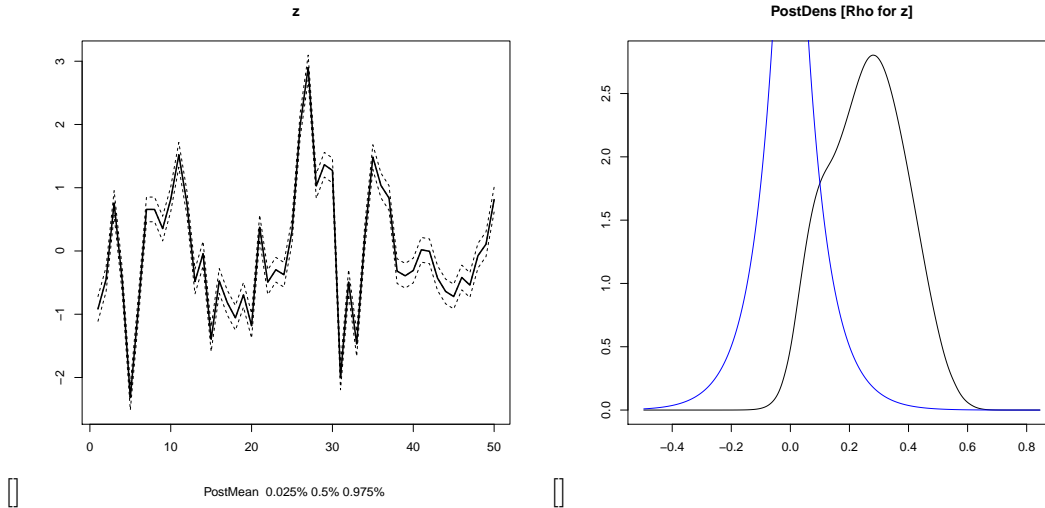


Figure 3: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.4$  and  $\alpha = 0.005$

### Case 2 : Effect of user defined scale when PC priors are derived assuming base model to be No change in time ( $\phi = 1$ ):

We have simulated a time series data of an AR(1) process, using the  $\phi = 0.95$ . Tried to fit it using various user defined scale to see the effect of user defined scales on priors, we have considered three user defined scales, please note that the probability statement in this case is as follows

$$Prob(|\phi| > U) = \alpha,$$

we have defined three scales for  $U = .95$  and  $\alpha = (0.5, 0.7, 0.9)$ :

```
#Case 2 : for different values of use defined
###scale when base model is considered with model rho =1

n1 = 50
rho1 = 0.95
s1 = 0.1

#simulate a AR(1) data for a given marginal precision and rho
simulated_ts1 = scale(as.numeric(arima.sim(n = n1, list(ar=c(rho1)))))+ rnorm(n,
                                                                              sd=s1)

# define values for a suitable guess
u1 = .95
alpha1 = c(0.5 ,0.7, 0.9)

# define prior distribution for hyper parameters
# define data frame as a input to inla
data1 = list(y=simulated_ts1, z=1:n1)

#for (i in 1:3) {

formula11 = simulated_ts1~ -1 + f(z,model="ar1",
                                hyper = list(rho = list(prior = "pc.cor1",
                                                         param = c(u1, alpha1[1])),
                                prec = list(initial = log(1), fixed=TRUE)))

formula12 = simulated_ts1~ -1 + f(z,model="ar1",
```

```

hyper = list(rho = list(prior = "pc.cor1",
                        param = c(u1, alpha1[2])),
            prec = list(initial = log(1), fixed=TRUE)))
formula13= simulated_ts1~ -1 + f(z,model="ar1",
            hyper = list(rho = list(prior = "pc.cor1",
                                    param = c(u1, alpha1[3])),
                        prec = list(initial = log(1), fixed=TRUE)))

r11 = inla (formula11, data = data1,
            control.family = list(hyper = list(
                prec = list(initial = log(1/s1^2), fixed = TRUE))))

r12 = inla (formula12, data = data1,
            control.family = list(hyper = list(
                prec = list(initial = log(1/s1^2), fixed = TRUE))))

r13 = inla (formula13, data = data1,
            control.family = list(hyper = list(
                prec = list(initial = log(1/s1^2), fixed = TRUE))))
#par(mfrow = c(3,2))
r11=inla.hyperpar(r11)
r12=inla.hyperpar(r12)
r13=inla.hyperpar(r13)

```

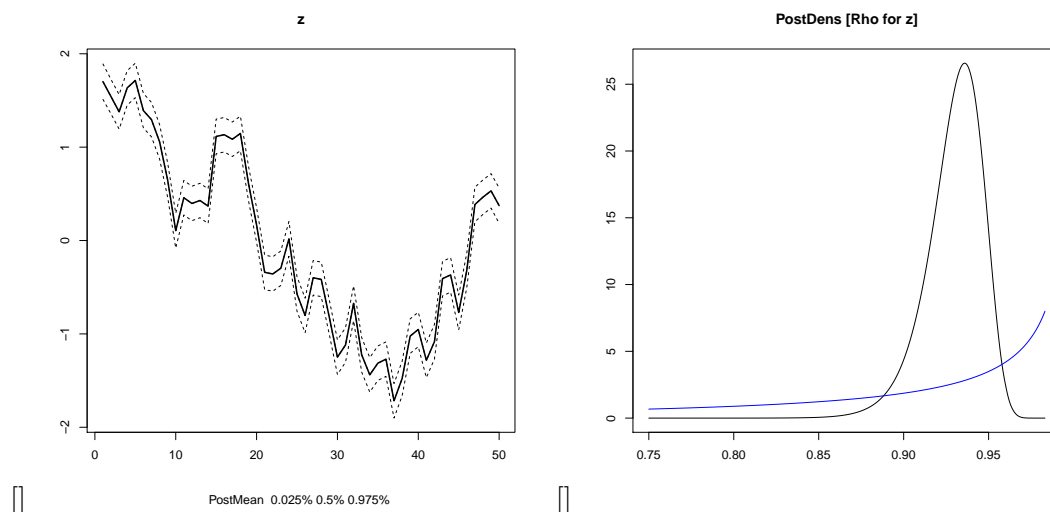


Figure 4: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.95$  and  $\alpha = 0.5$

We can see from Figure 4 that probability mass for  $\phi$  is concentrated between .9 and .95, we have put mild conditions on prior, we have mentioned in this case that only 50 % probability mass can stay below the  $U = .95$ , We have got following estimates in this case

```

r11$summary.hyperpar
##              mean          sd 0.025quant  0.5quant 0.975quant      mode
## Rho for z 0.9297177 0.01657332 0.8911925 0.9318892 0.9557419 0.9358997

```

In Figure 5, we have considered a more strong user defined scale for PC Priors, We have set that only 30% probability mass pf  $\phi$  can stay below the 0.95. So the estimation for  $\phi$  is closer to base model than in the case of Figure 4. We have got following estimates in this case

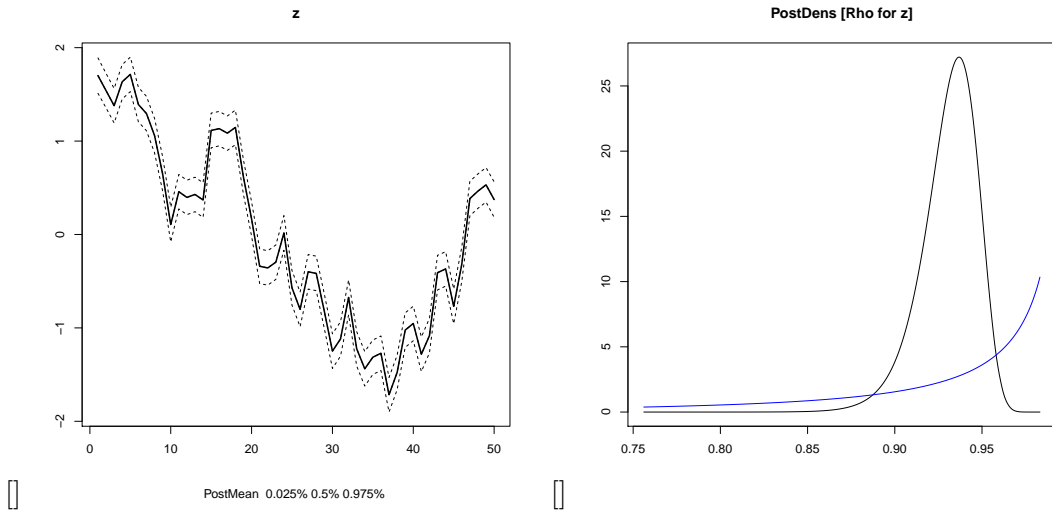


Figure 5: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.95$  and  $\alpha = 0.7$

```
r12$summary.hyperpar
```

##	mean	sd	0.025quant	0.5quant	0.975quant	mode
## Rho for z	0.9308732	0.01614846	0.8934059	0.9329654	0.9563056	0.9368381

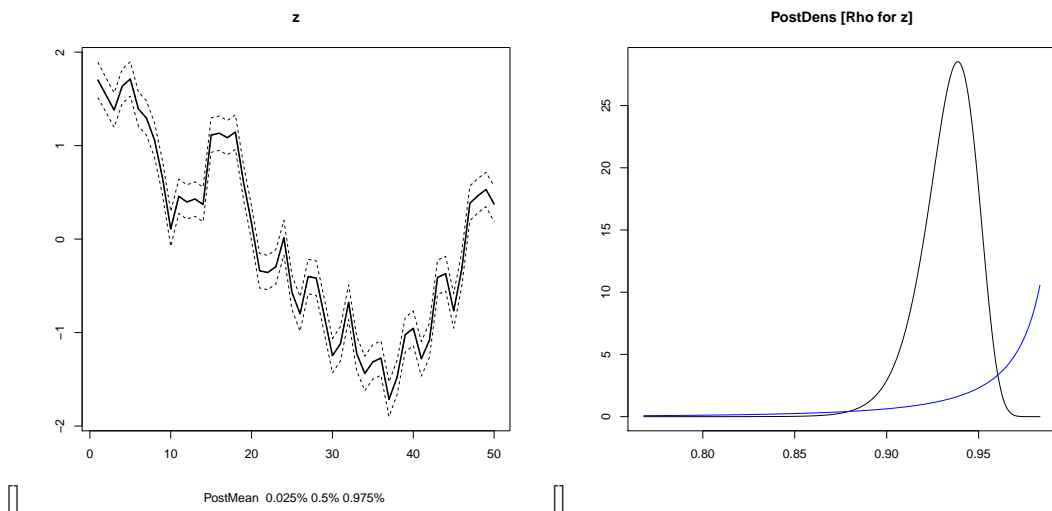


Figure 6: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.95$  and  $\alpha = 0.9$

In Figure 6, we have considered a strong user defined scale for PC Priors, We have set that only 10% probability mass of  $\phi$  can stay below the 0.95. So the estimation for  $\phi$  is closer to the base model. We have got following estimates in this case

```
r13$summary.hyperpar
```

##	mean	sd	0.025quant	0.5quant	0.975quant	mode
## Rho for z	0.9331499	0.01532927	0.897707	0.935092	0.957431	0.9387023

We observe that even when we considered the different base model for constructing PC priors the effect of user defined scale remains the same. So, the better user defined scale we have for our data, priors become more informative about the data. We also observe that PC priors shows excellent shrinkage towards the base model. The amount of shrinkage will depend on the user defined scales.



### Case 3: Effect of changing error variance on estimation using PC Priors considering the Base Model: $\phi = 0, x_1 \sim \mathcal{N}(0, 1)$ :

We have seen the effect of user defined scale on the construction of PC Priors, Now we will see the effect of changing the variance of error terms in time series data on the PC Priors, Now we will study the effect of changing error variance on PC Priors and estimation of parameter, when we have constructed PC Priors using the base model  $\phi = 0$ . We have simulated a AR(1) data using  $\phi = 0.4$ . We have defined three scale for error terms standard deviation ie 0.1, 0.5, 1 under this case, we have considered following user defined scale for construction of PC Priors  $U=0.4$  and  $\alpha = 0.5$  ie 50% of probability mass can remain inside the maximum value of 0.4:

```
#CASE 3

n = 50
rho3 = 0.4
s3 = c(0.1, 0.5, 1)
#simulate a AR(1) data for a given marginal precision and rho

ts30 = scale(as.numeric(arima.sim(n = n, list(ar=c(rho3))))))

simulated_ts31 = ts30 + rnorm(n, sd=s3[1])
simulated_ts32 = ts30 + rnorm(n, sd=s3[2])
simulated_ts33 = ts30 + rnorm(n, sd=s3[3])

# define values for a suitable guess
u3 = .4
alpha3 = 0.5

# define prior distribution for hyper parameters
# define data frame as a input to inla
data31 = list(y=simulated_ts31, z=1:n)
data32 = list(y=simulated_ts32, z=1:n)
data33 = list(y=simulated_ts33, z=1:n)

#for (i in 1:3) {

formula31 = simulated_ts31 ~ -1 + f(z,model="ar1",
                                hyper = list(rho = list(prior = "pc.cor0",
                                                         param = c(u3, alpha3)),
                                              prec = list(initial = log(1), fixed=TRUE)))

formula32 = simulated_ts32 ~ -1 + f(z,model="ar1",
                                hyper = list(rho = list(prior = "pc.cor0",
                                                         param = c(u3, alpha3)),
                                              prec = list(initial = log(1), fixed=TRUE)))

formula33 = simulated_ts33 ~ -1 + f(z,model="ar1",
                                hyper = list(rho = list(prior = "pc.cor0",
                                                         param = c(u3, alpha3)),
                                              prec = list(initial = log(1), fixed=TRUE)))

r31 = inla (formula31, data = data31,
            control.family = list(hyper = list(
                prec = list(initial = log(1/s3[1]^2), fixed = TRUE))))
```

```

r32 = inla (formula32, data = data32,
           control.family = list(hyper = list(
             prec = list(initial = log(1/s3[2]^2), fixed = TRUE))))

r33 = inla (formula33, data = data33,
           control.family = list(hyper = list(
             prec = list(initial = log(1/s3[3]^2), fixed = TRUE))))

r31=inla.hyperpar(r31)
r32=inla.hyperpar(r32)
r33=inla.hyperpar(r33)

```

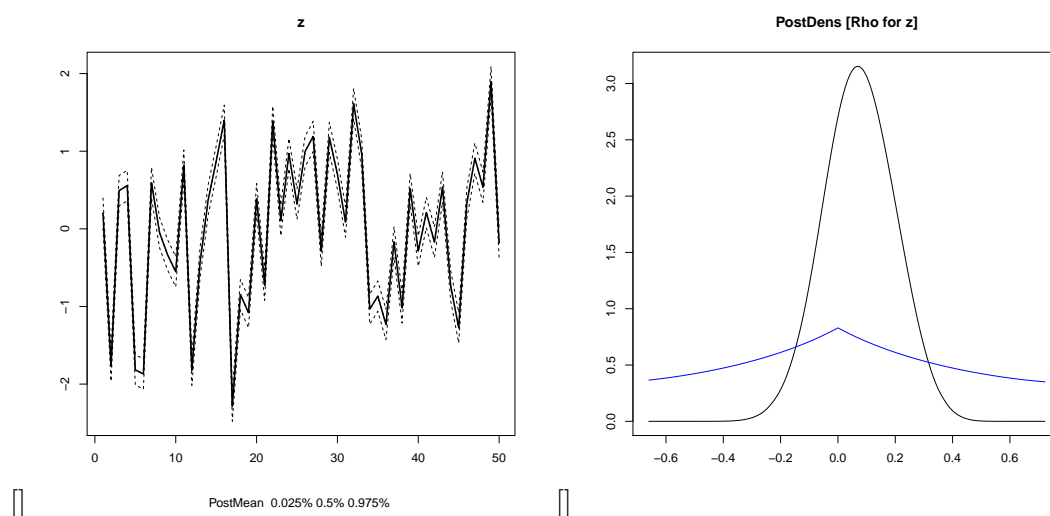


Figure 7: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.4$  and  $\alpha = 0.5$

In this case error standard deviation is 0.1 so posterior for  $\phi$  in the Figure 7 looks like Gaussian curve, since our PC priors behaves almost like a flat prior, so estimation of  $\phi$  will depends on the likelihood of data. In this case we got following results:

```

r31$summary.hyperpar

##              mean          sd 0.025quant  0.5quant 0.975quant      mode
## Rho for z 0.07459386 0.1245771 -0.169116 0.07393633 0.3170877 0.06928596

```

In this case error standard deviation is 0.5 so posterior for  $\phi$  in the Figure 8 looks like Gaussian but it has more spread than the Figure 7. Increasing error variance has broadened the spread of posterior. Estimation of  $\phi$  would be around mode of posterior as since our PC priors behaves almost like a flat prior, so estimation of  $\phi$  will depends on the likelihood of data. In this case we got following results:

```

r32$summary.hyperpar

##              mean          sd 0.025quant  0.5quant 0.975quant      mode
## Rho for z 0.1425958 0.1402965 -0.1289104 0.1410096 0.4161577 0.1299892

```

In this case error standard deviation is 1 so posterior for  $\phi$  in the Figure 9 looks like distorted Gaussian but it has maximum spread wrt to the Figure 7, Figure 8. Increasing error variance has broadened the spread of posterior, so estimation of  $\phi$  would be around mode of posterior as our PC priors behaves almost like a flat prior, so estimation of  $\phi$  will depends on the likelihood of data. In this case we got following results:

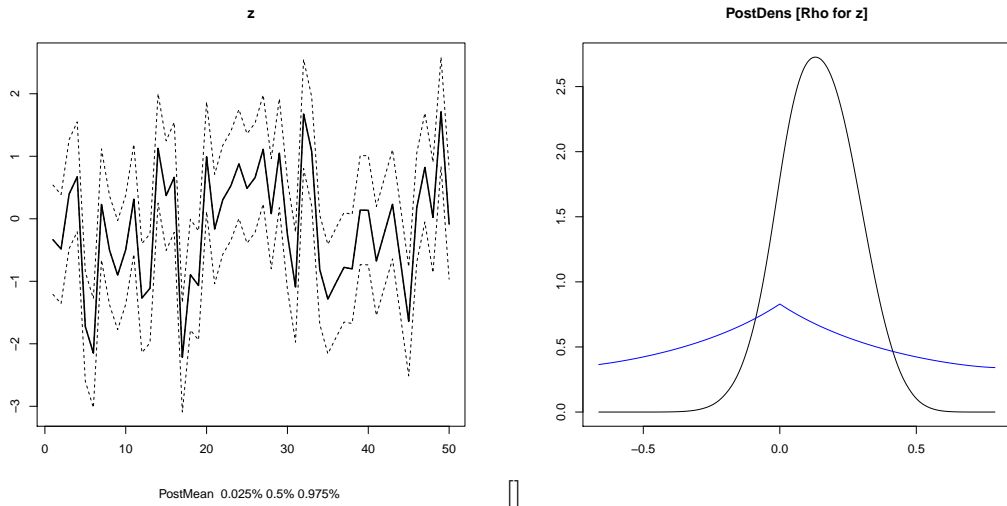


Figure 8: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.4$  and  $\alpha = 0.5$

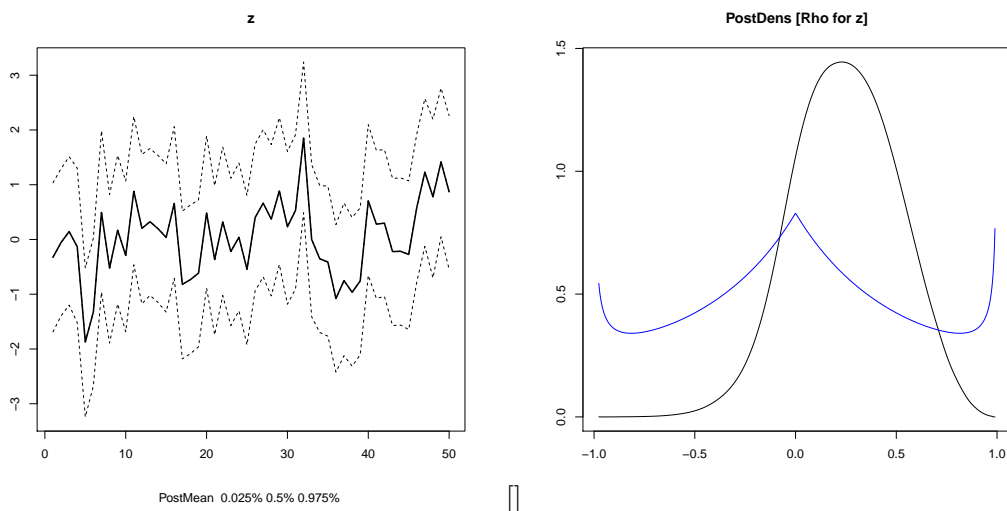


Figure 9: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.4$  and  $\alpha = 0.5$

```
r33$summary.hyperpar
```

```
##           mean           sd 0.025quant  0.5quant  0.975quant    mode
## Rho for z 0.2471002 0.2554012 -0.2512596 0.2465804  0.729711  0.22954
```

We observe that changing error variance certainly doesn't affect the construction of PC Priors but it has a great effect on the estimation of parameter. As we see that PC Prior has the same shape for all three cases. But estimation of  $\phi$  changed as likelihood of data got affected by change of variance of error term. Likelihood becomes the key factor in this case as PC Prior behaves like a flat prior with user defined setting in this case. Estimation of  $\phi$  is certainly more close to the mode of posterior as maximum probability mass is centered around mode of posterior.

#### Case 4: Effect of changing error variance on estimation using PC Priors considering the Base Model: $\phi = 1, x_1 \sim \mathcal{N}(0, 1)$ :

In this section, we will see the effect of adding noise to our data on estimation on the parameter of interest using the PC Prior constructed using  $\phi = 1$  for the base model. We have simulated a AR(1) data using  $\phi = 0.95$ . We have defined three scale for error terms standard deviation ie 0.1, 0.5, 1 under this case, we have considered very mild user defined scale for construction of PC Priors  $U=0.95$  and  $\alpha = 0.5$  ie 50% of probability mass can remain inside the maximum value of 0.95:

```

n = 50
rho4 = 0.95
s3 = c(0.1, 0.5, 1)
#simulate a AR(1) data for a given marginal precision and rho

ts40=scale(as.numeric(arima.sim(n = n, list(ar=c(rho4))))))

simulated_ts41 = ts40 + rnorm(n, sd=s3[1])
simulated_ts42 = ts40 + rnorm(n, sd=s3[2])
simulated_ts43 = ts40 + rnorm(n, sd=s3[3])

# define values for a suitable guess
u4 = .95
alpha4 = 0.50

# define prior distribution for hyper parameters
# define data frame as a input to inla
data41 = list(y=simulated_ts41, z=1:n)
data42 = list(y=simulated_ts42, z=1:n)
data43 = list(y=simulated_ts43, z=1:n)

#for (i in 1:3) {

formula41 = simulated_ts41 ~ -1 + f(z,model="ar1",
                                hyper = list(rho = list(prior = "pc.cor1",
                                                         param = c(u4, alpha4)),
                                              prec = list(initial = log(1), fixed=TRUE)))

formula42 = simulated_ts42 ~ -1 + f(z,model="ar1",
                                hyper = list(rho = list(prior = "pc.cor1",
                                                         param = c(u4, alpha4)),
                                              prec = list(initial = log(1), fixed=TRUE)))

formula43 = simulated_ts43 ~ -1 + f(z,model="ar1",
                                hyper = list(rho = list(prior = "pc.cor1",
                                                         param = c(u4, alpha4)),
                                              prec = list(initial = log(1), fixed=TRUE)))

r41 = inla (formula41, data = data41,
           control.family = list(hyper = list(
                                 prec = list(initial = log(1/s3[1]^2), fixed = TRUE))))

r42 = inla (formula42, data = data42,
           control.family = list(hyper = list(
                                 prec = list(initial = log(1/s3[2]^2), fixed = TRUE))))

r43 = inla (formula43, data = data43,
           control.family = list(hyper = list(
                                 prec = list(initial = log(1/s3[3]^2), fixed = TRUE))))
r41=inla.hyperpar(r41)
r42=inla.hyperpar(r42)

```

```
r43=inla.hyperpar(r43)
```

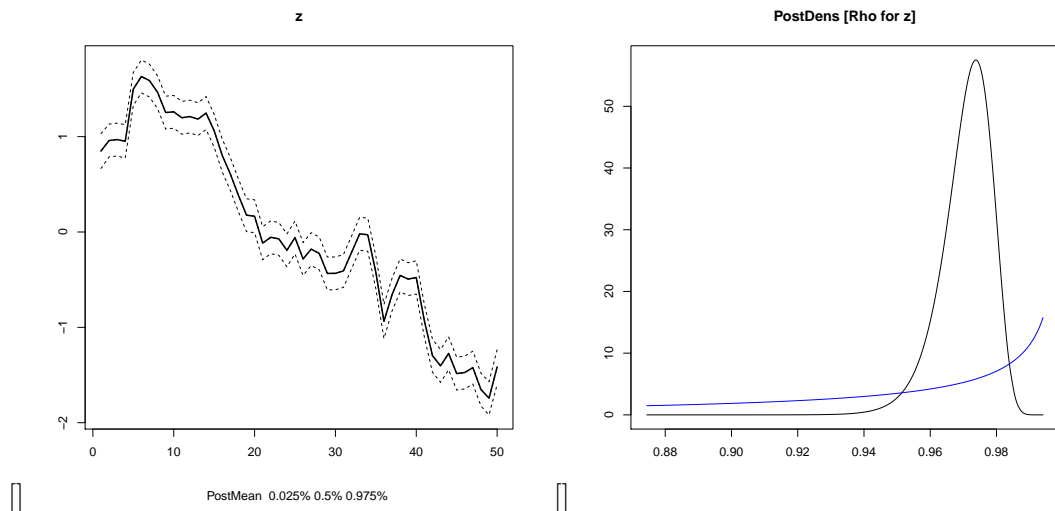


Figure 10: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.95$  and  $\alpha = 0.5$

In this case error standard deviation is 0.1 so posterior for  $\phi$  in the Figure 10 looks nice Gaussian curve which has mode between 0.8 and 0.9, since our PC priors behaves like weakly informative prior in this user defined scale, so estimation of  $\phi$  will depends on the likelihood of data. In this case we got following results:

```
r41$summary.hyperpar
##           mean          sd 0.025quant  0.5quant 0.975quant    mode
## Rho for z 0.9707873 0.007713815  0.9527703 0.9718371  0.982767 0.973772
```

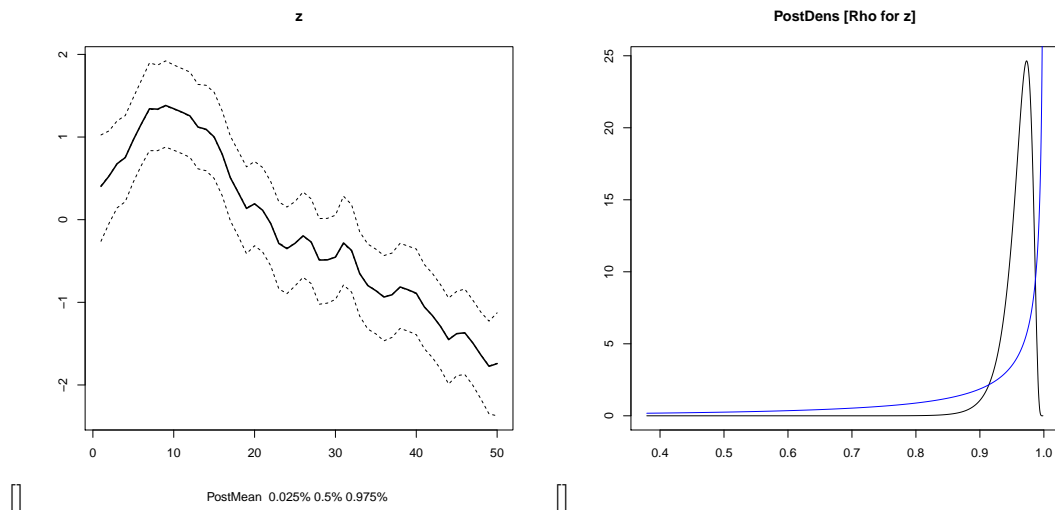


Figure 11: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.94$  and  $\alpha = 0.5$

In this case error standard deviation is 0.5 so posterior for  $\phi$  in the Figure 11 looks Gaussian but it has more spread than the Figure 10. Increasing error variance has broaden the spread of posterior, however mod of posterior has changed from where it was in Figure 10. Estimate of  $\phi$  are around mode of posterior, since our PC priors behaves like weakly informative prior with this user defined scale, so estimation of  $\phi$  will depends on the likelihood of data. In this case we got following results:

```
r42$summary.hyperpar
##           mean          sd 0.025quant  0.5quant 0.975quant    mode
## Rho for z 0.9593106 0.02221302  0.9024545 0.9643524  0.9870136 0.9729796
```

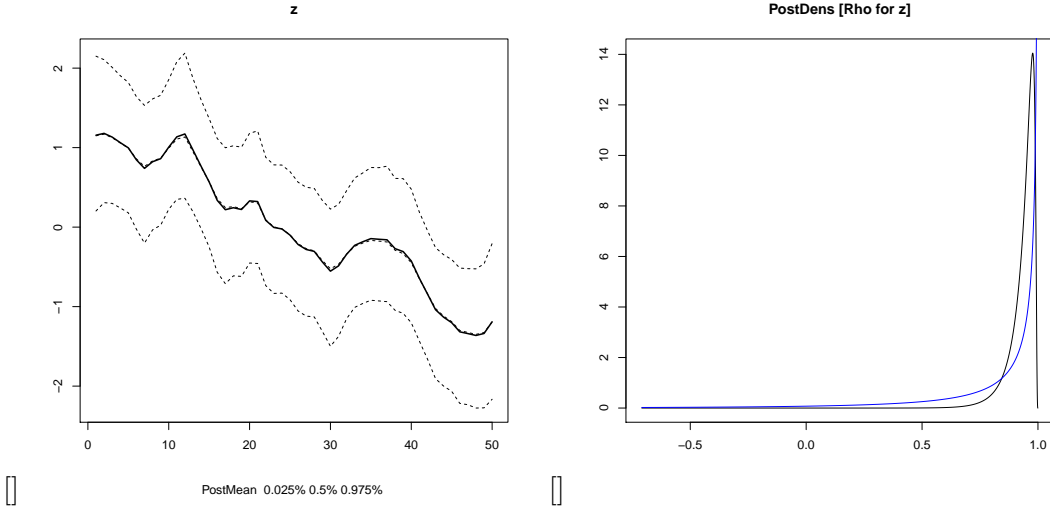


Figure 12: Plot of PC Priors(in Blue) when user defined scales when  $U = 0.95$  and  $\alpha = 0.5$

In this case error standard deviation is 1 so posterior for  $\phi$  in the Figure 12 looks like skewed Gaussian but it has maximum spread wrt to the Figure 10, Figure 11. Increasing error variance has broadened the spread of posterior, however mode of posterior is shifted when compared to earlier 2 cases, so estimates of  $\phi$  would be around mode of posterior, since our PC priors behaves like a weakly informative prior, so estimation of  $\phi$  will depends on the likelihood of data. In this case we got following results:

```
r43$summary.hyperpar
```

##	mean	sd	0.025quant	0.5quant	0.975quant	mode
## Rho for z	0.9355549	0.05505409	0.7856487	0.952011	0.9908003	0.9774016

We conclude this section by observing that changing error variance certainly doesn't affect the construction of PC priors but it has a great effect on the estimation of parameter. As we see that PC Prior has the same shape for all three cases. But estimation of  $\phi$  changed as likelihood of data got affected by change of variance of error term. Likelihood becomes the key factor in the case where prior behaves like a flat prior. In this case, PC prior behaves like flat prior as we have provided weak user defined scales for PC priors. Estimation of  $\phi$  is certainly more close to the mode of posterior as maximum probability mass is centered around the mode of posterior.

## 7 Comparison of PC Priors with Reference and Jeffery's Priors

In Bayesian setting, some time priors are preferred that doesn't strongly influence the posterior distribution. These priors are called uninformative priors. We can assign uniform prior to our hyper parameter but that may not be invariant towards reparameterization. So Jeffery's priors are the generalization of the idea of uninformative priors. And Jeffery's priors are based on principle of invariance. Jeffery's prior are calculated by following formula [3] [4.6.2 Priors and Posteriors in the presence of a unit root]jeffreys priors for AR(1) :

$$\pi_J(\theta) = \sqrt{\det(I(\theta))}$$

where  $I_{ij}(\theta) = \text{Fisher Information matrix} = -E_{\theta}(\frac{\partial^2 \log(pX|\theta)}{\partial \theta_i \partial \theta_j})$ ,  
in our case of AR(1) process,

$$\det(I(\theta)) = (\{\frac{n}{1-\phi^2} + \frac{1-\phi^{2n}}{1-\phi^2} \{E(\frac{X_0^2}{\sigma^2}) + \frac{1}{1-\phi^2}\}\})$$

which gives us  $\pi(\phi) \propto \frac{(1-\phi)^2}{\sigma^2} \sqrt{\frac{1}{1-\phi^2} (n + \frac{1-\phi^{2n}}{1-\phi^2})}$ . which is un-normalized, which simplifies to following for large values of n:

$$\pi(\phi) \propto \sqrt{\frac{1-\phi}{1+\phi}}(1-\phi)$$

We would like to construct prior distribution for the notion of distance using the Jeffery's prior distribution. We have two notion of the distance depending on our base model. So in the first case when we considered Base model with  $\phi = 0$ , we have got  $d(\phi) = \sqrt{(1-p) \log(1-\phi^2)}$  which gives us

$$\phi = \sqrt{1 - \exp\left(-\frac{d^2}{n-1}\right)}$$

So we can find prior distribution for notion of distance using the following:

$$\pi_J(d) = \pi_J(\phi) \left| \frac{\partial(\phi)}{\partial d} \right|$$

which gives

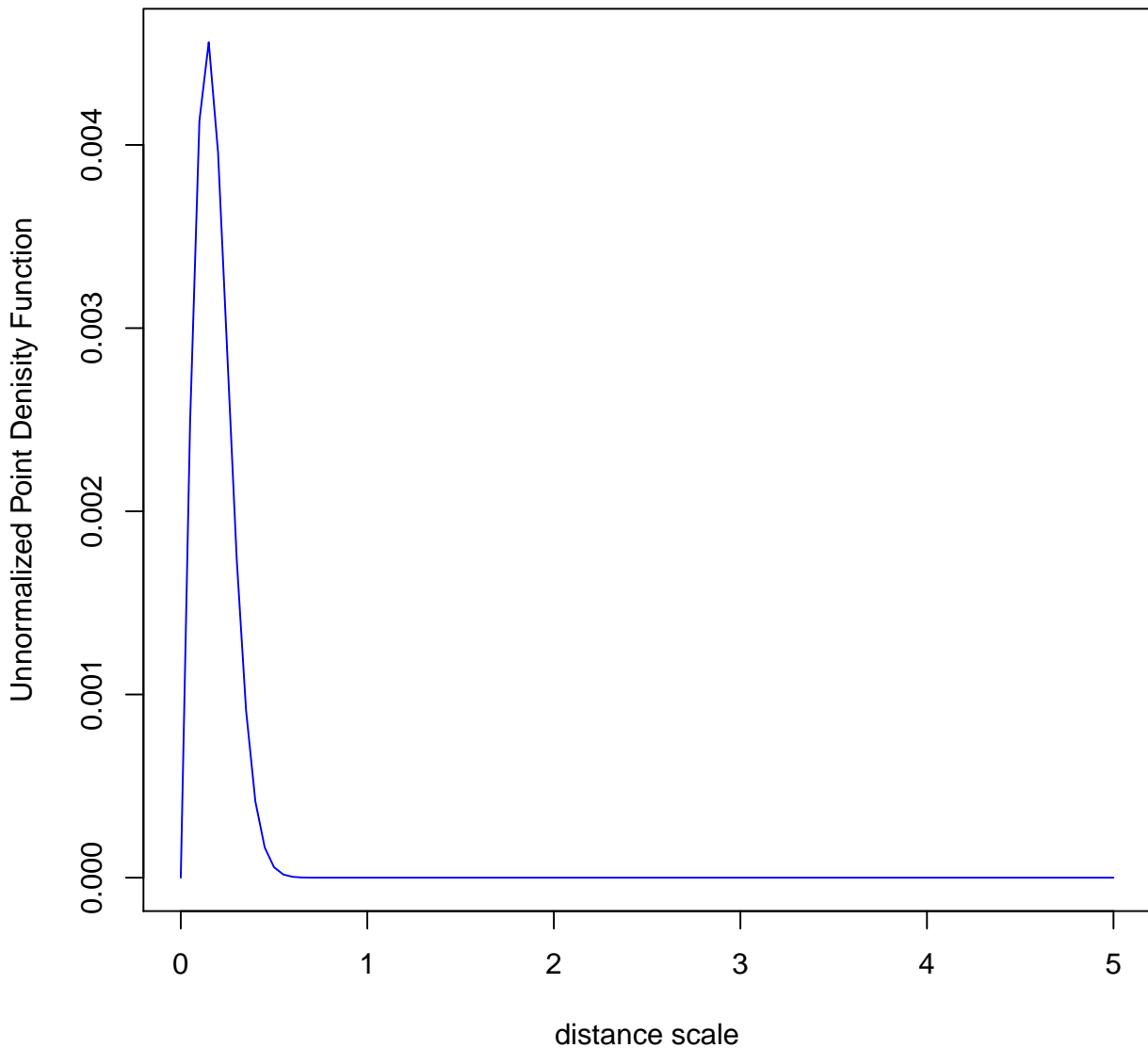
$$\pi_J(d) \propto \frac{\exp\left(\frac{-d^2}{2(n-1)}\right) \left(1 - \sqrt{1 - \exp\left(\frac{-d^2}{(n-1)}\right)}\right)^2(d)}{\sqrt{1 - \exp\left(\frac{-d^2}{(n-1)}\right)}}$$

If we plot this prior the shape of the prior looks like following:

```
curve((exp(-x^2/2*(50-1)))*(1-sqrt(1-exp(-x^2/50-1)))^2*x/sqrt(1-exp(-x^2/50-1)),
      from = 0, to = 5, xlab = "distance scale",
      ylab = "Unnormalized Point Denisity Function",
      col = "blue")

title(main = "Jeffrey Prior for distance when distance calculated
           considering phi =0 the base model")
```

## Jeffrey Prior for distance when distance calculated considering phi =0 the base model



Now, we will construct prior distribution for the case when we considered Base model with  $\phi = 1$ , we have got  $d(\phi) = c\sqrt{1-\phi}$ . which gives us

$$\phi = 1 + \frac{-d^2}{c}$$

for the sake of simplicity we assume that  $c = 1$ . Which will put a constraints on  $d$  that  $d \leq 1$ . So we can find prior distribution for notion of distance using the following:

$$\pi_J(d) = \pi_J(\phi) \left| \frac{\partial(\phi)}{\partial d} \right|$$

which gives

$$\pi_J(d) \propto \frac{d^4}{\sqrt{(2-d^2)}}$$

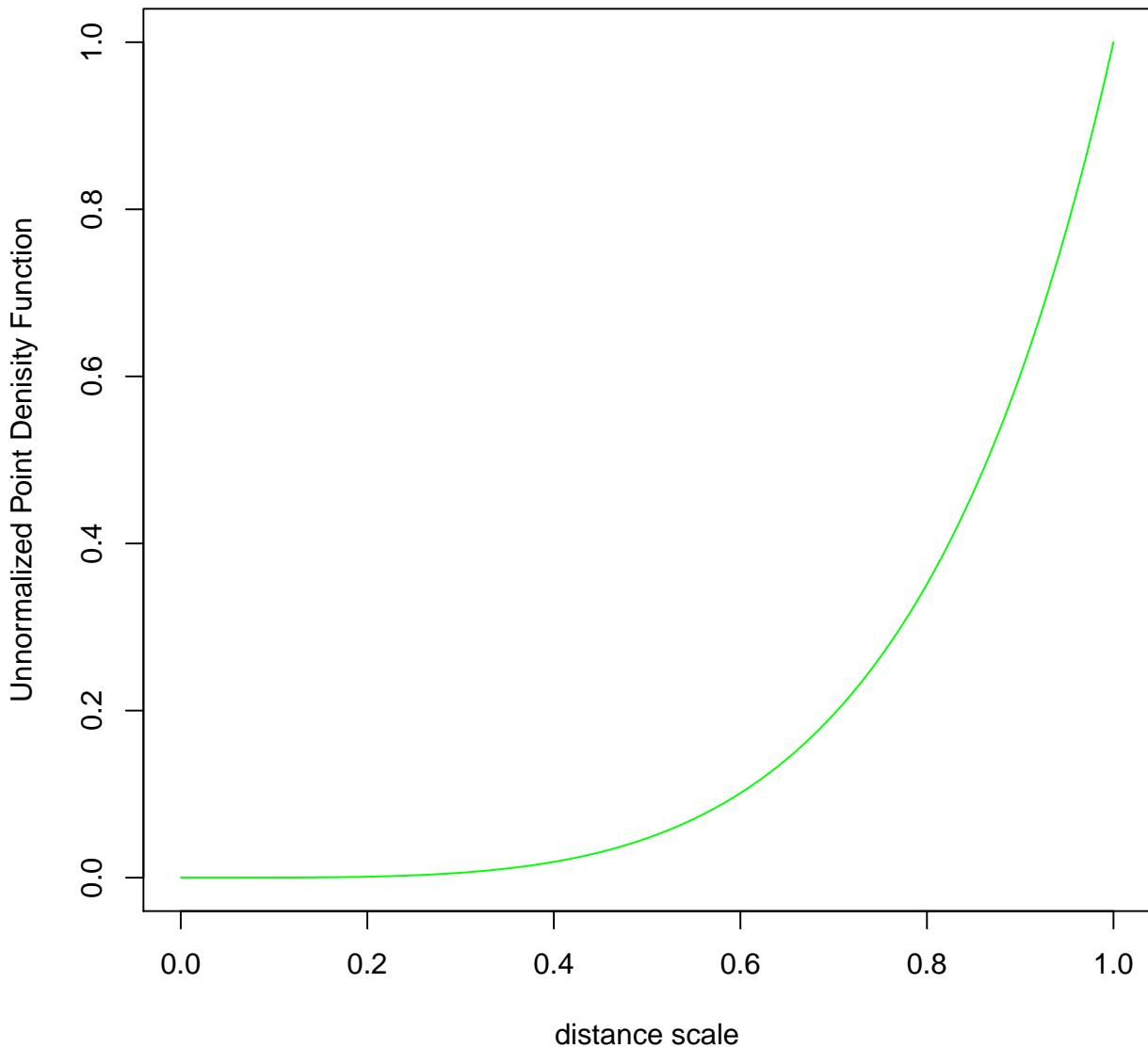
If we plot this prior this looks like following:

```
curve((x^4)/sqrt(2-x^2), from = 0, to =1,
      xlab = "distance scale", ylab = "Unnormalized Point Denisity Function",
      col= "Green")

title(main = "Jeffrey Prior for distance when distance calculated
            considering phi =1 the base model")
```



## Jeffrey Prior for distance when distance calculated considering phi =1 the base model



We can observe that for both choices of base model, pdf for notion of distance overfits ie it give zero probability mass for cases when d is close to 0. Which contradicts the assumptions of Occams Razor. If we look at the pdf curves, they make it more probable for some distances which are not close to base model. Jeffrey priors are often used for defining the prior distribution for parameters for AR(1) process. But origin of Jeffreys prior is to give uninformative priors which are invariant under reparameterization. However, if we look at them with the perspective of distance from the base model, they don't support the idea that base model is more probable. They suggest that models which are closer to base model are less probable than some particular distances.

Now, we will construct prior distribution for the notion of distance considering the reference prior distribution for our parameter of model ie  $\phi$ . The idea behind reference priors is to formalize what exactly we mean by an "uninformative prior". It is a function that maximizes some measure of distance or divergence between the posterior and prior, as data observations are made. Reference Prior for correlation at lag 1 of an AR(1) process are to be given by following pdf[4]:

$$\pi_R(\phi) = \frac{1}{\pi} \frac{1}{\sqrt{(1 - \phi^2)}}$$

We would like to construct prior distribution for the notion of distance using the Reference prior distribution. We have two notion of the distance depending on our base model. So in the first case

when we considered Base model with  $\phi = 0$ , we have got  $d(\phi) = \sqrt{(1-p) \log(1-\phi^2)}$  which gives us

$$\phi = \sqrt{1 - \exp\left(-\frac{d^2}{n-1}\right)}$$

So we can find prior distribution for notion of distance using the following:

$$\pi_R(d) = \pi_R(\phi) \left| \frac{\partial(\phi)}{\partial d} \right|$$

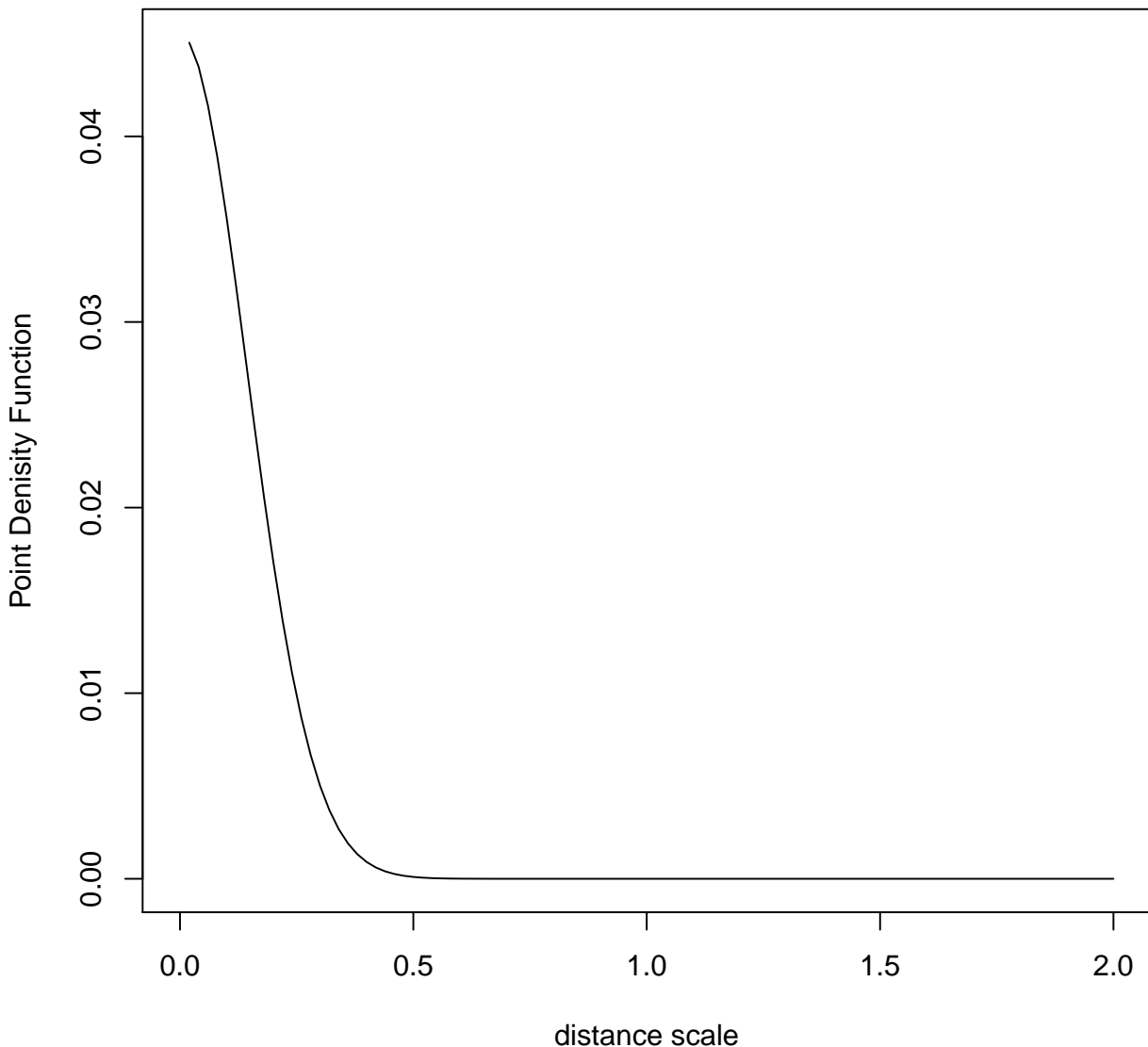
which gives

$$\pi_R(d) = \frac{1}{\pi} \frac{\exp\left(-\frac{d^2}{2(n-1)}\right) \left(1 - \exp\left(-\frac{d^2}{n-1}\right)\right)^{-0.5} d}{(n-1)}$$

If we plot this prior this looks like following:

```
curve((1/3.14)* (exp( -x^2/2*(50-1))) / (1-exp( -x^2/(50-1)))^(.5) * (x / (50-1)),
      from = 0, to = 2, xlab = "distance scale", ylab = "Point Denisity Function")
title(main = "Reference Prior for distance when distance calculated
          considering phi =0 the base model")
```

**Reference Prior for distance when distance calculated  
considering phi =0 the base model**



Now, we will construct prior distribution for the case when we considered Base model with  $\phi = 1$ , we have got  $d(\phi) = c\sqrt{1 - \phi}$ . which gives us

$$\phi = 1 + \frac{-d^2}{c}$$

for the sake of simplicity we assume that  $c = 1$ . Which will put a constraints on  $d$  that  $d \leq 1$ . So we can find prior distribution for notion of distance using the following:

$$\pi_R(d) = \pi_R(\phi) \left| \frac{\partial(\phi)}{\partial d} \right|$$

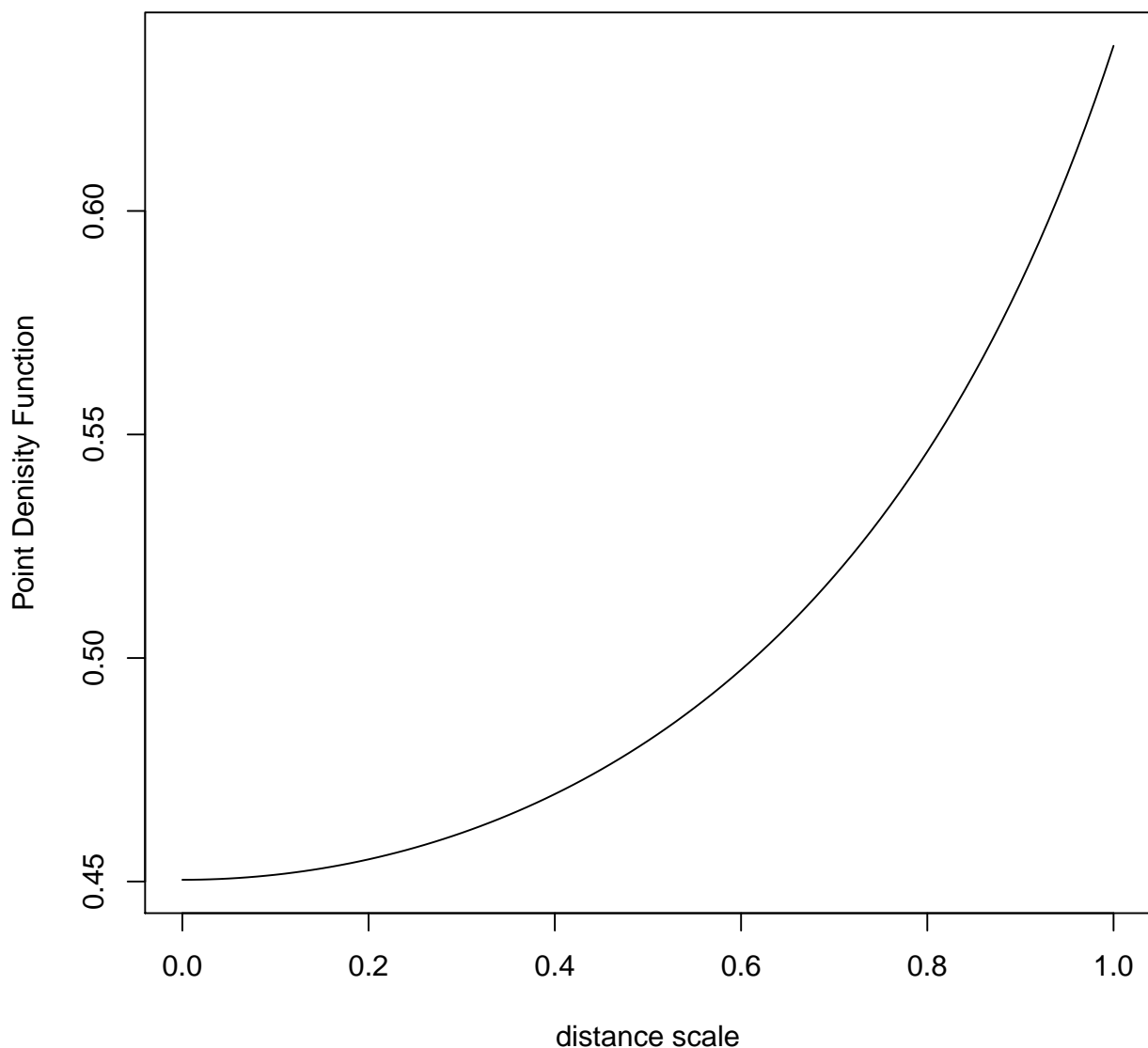
which gives

$$\pi_R(d) = \frac{2}{3.14} \frac{1}{\sqrt{2 - d^2}}$$

If we plot this prior this looks like following:

```
curve((2/3.14)/ (2-x^2)^.5, from = 0, to = 1,
      xlab = "distance scale", ylab = "Point Density Function")
title(main = "Reference Prior for distance when distance calculated
           considering phi =1 the base model")
```

### Reference Prior for distance when distance calculated considering phi =1 the base model



We can observe that reference prior looks reasonable when we consider distance from the base model corresponds to  $\phi = 0$ , as it doesn't overfit distances closes to 0. It supports the idea that model close to base model are more probable than the model far from base model. However, if we choose alternative base model ie  $\phi = 1$ , reference prior doesn't look that reasonable as it makes probable those model which are far from base model. Clearly, when the  $\phi$  is close to 0 for our data set Reference prior may be used. But  $\phi$  is close to 1, reference prior approach may not be suitable. PC Priors are based on constant penalization principle hence they support both the base models and can be used for any general or specific setting of  $\phi$ .

## 8 Conclusions/Discussions

Prior selection is the crucial issue in the Bayesian setting, priors are the strength of Bayesian statistical inference at the same time the method of prior selections becomes a point of critical discussions. In this paper, we have tried to explain a systematic approach to construct prior for parameter of interest. We construct prior for every hyper parameter, and they are based on the selection of the base model. So the most important part of whole construction is to found a suitable base model for the given problem. There can be several approaches to select base model, based on the selection of base model the prior distribution for the hyperparameter will change. We will construct priors for MA(1) processes, then ARMA(1,1) and further to MA(q) time series models following the same approach which has been presented in this report.

## 9 Acknowledgement

I would like to express my deepest thanks to Professor Håvard rue, for providing me opportunity to work with him and for his motivational and educational supervision.

## References

- [1] Daniel P. Simpsons Thiago G. Martins Andrea Riebler Geir-Arne Fugstad Håvard Rue Sigrunn H. Sørbye Penalising model component complexity: A principled, practical approach to construct priors (December 1, 2014).
- [2] Sigrunn H. Sørbye Håvard Rue Penalized complexity priors for stationary autoregressive processes (August 29, 2016)
- [3] Book by In Choi Almost All About Unit Roots: Foundations, Developments, and Applications page 108
- [4] James O. Berger Ruo-young Yang Noninformative Priors and Bayesian Testing for AR(1) Model Technical Report 92-45C, page 7