## Example2abstraction

### Original

```
## Assign spatial coordinates
coordinates(sp1) <- ~decimalLongitude + decimalLatitude ## Transform occurrences to spdataframe
proj4string(sp1) <- CRS("+proj=longlat +datum=WGS84")
sp1 <- spTransform(sp1, crs(climateRasters))

## Generate sample area for background points
samplearea <- buffer(sp1, width = 100000, dissolve = T) ## 100 km buffer
crs(samplearea) <- crs(sp1)

## determine the number of background points
nbackgr <- ifelse(nrow(sp1) > 9999, nrow(data.frame(sp1)) ,10000) ## 10k unless occurrences are more th

## Restrict background points to reduce bias
samplearea <- mask(climateRasters[[1]], samplearea)
backgr <- randomPoints(samplearea, n=nbackgr, p =sp1)  %>% data.frame() ## sample points, exclude cells
coordinates(backgr) <- ~x+y ## re-assign as spatial points
proj4string(backgr)  <- crs(sp1) ## assign CRS

## Extract climate data
pres <- extract(climateRasters, sp1)
abs <- extract(climateRasters, backgr)
allClim <- rbind(pres, abs) %>% data.frame()

## Check for covariance
colin <- usdm::vifcor(allClim[,-ncol(allClim)])
selectVars <-  colin@results$Variables

## Use best variables without colinearity issues
bestClim <- climateRasters[[selectVars]]
```

### With Abstraction

```
sp1 <- assignSpatialCoordinates(sp1)

## Generate sample area for background points
samplearea <-BufferAroundPoints(sp1, 100000)

## Restrict background points to reduce bias
nbackgr <- getNumberBackgrPoints(sp1)
backgr <- getBackgrPoints(samplearea, nbackgr, sp1)

###### Reduce co-linearity between models
selectedVars <- findNonCollinearVariables(partionedData$trainingPresence,
  partionedData$trainingAbsence,
  climateRasters)
bestClimate <- climateRasters[[selectedVars]] ## Use best variables without colinearity issues
```

**Extreme abstraction**

```
selectedVars <- findNonCollinearVariables(sp1, climateRasters)
bestClimate <- climateRasters[[selectedVars]]
```