

Analisis Estadístico Vuela Libre

Zaira García - Luis Requeni - Celia Sifre

27/11/2022

INTRODUCCIÓN

Realizamos inicialmente una descriptiva general de nuestro banco de datos.

```
datos <- read.table(file = "FID.dat", header = T)
```

Contamos con 18 observaciones (*colony*), anotando para cada una de ellas el numero de visitantes esperado (*visitors*) y la densidad de gaviotas (*density*) en un momento determinado. Igualmente queda reflejada la distancia media a la que huyeron los distintos ejemplares estudiados al interactuar con un humano (*meanFID*).

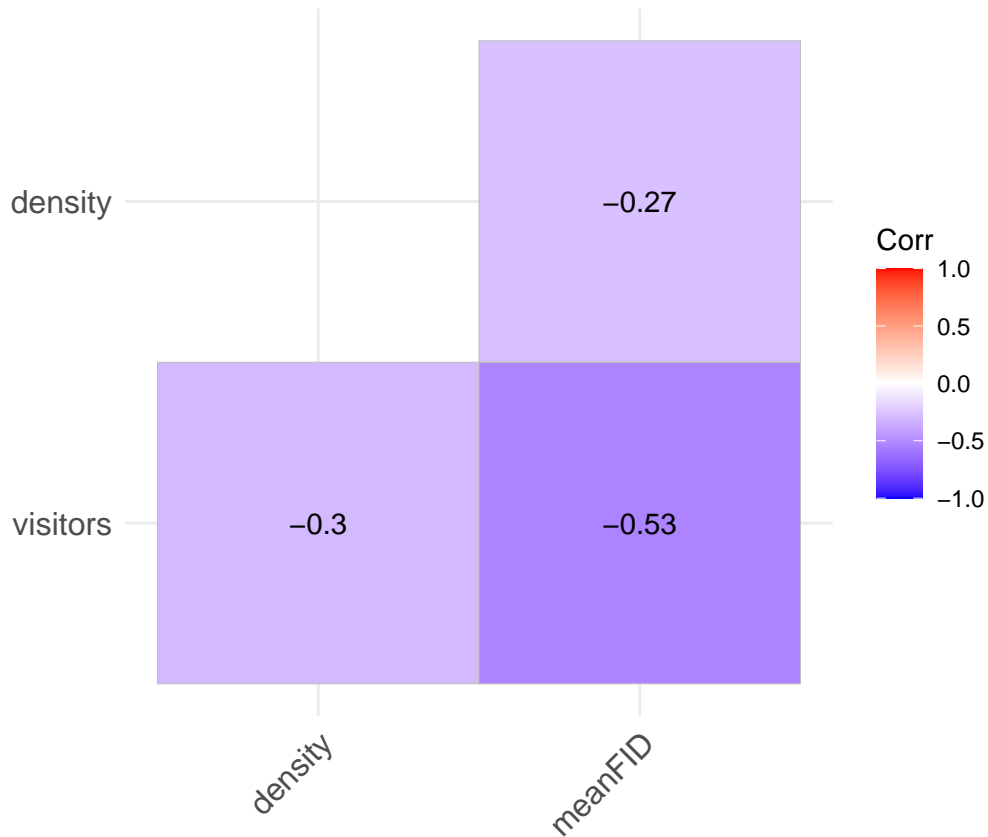
Un análisis descriptivo sencillo nos permite conocer someramente los detalles de nuestro banco de datos.

```
psych::describe(datos)
```

##	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
## colony	1	18	9.50	5.34	9.50	9.50	6.67	1.00	18.0	17.00	0.00
## visitors	2	18	1271.17	2288.46	2.00	964.56	0.00	2.00	7446.0	7444.00	1.45
## density	3	18	103.39	104.69	82.54	87.80	75.56	11.11	445.0	433.89	1.90
## meanFID	4	18	30.72	18.96	28.75	29.39	18.28	4.90	77.8	72.90	0.78
##	kurtosis	se									
## colony	-1.40	1.26									
## visitors	0.77	539.39									
## density	3.55	24.68									
## meanFID	-0.07	4.47									

Merece la pena destacar la alta desviación típica asociada a nuestra variable *visitors*. En 11 de nuestras 18 observaciones el número de visitantes es de tan solo 2 personas (investigadores). Mientras en que en 5 de estas observaciones los visitantes se miden en más de 1.000. Esta variabilidad nos hace pensar si habrá que trabajar con ella de forma numérica o recategorizarla según una afluencia de visitantes alta, media o baja.

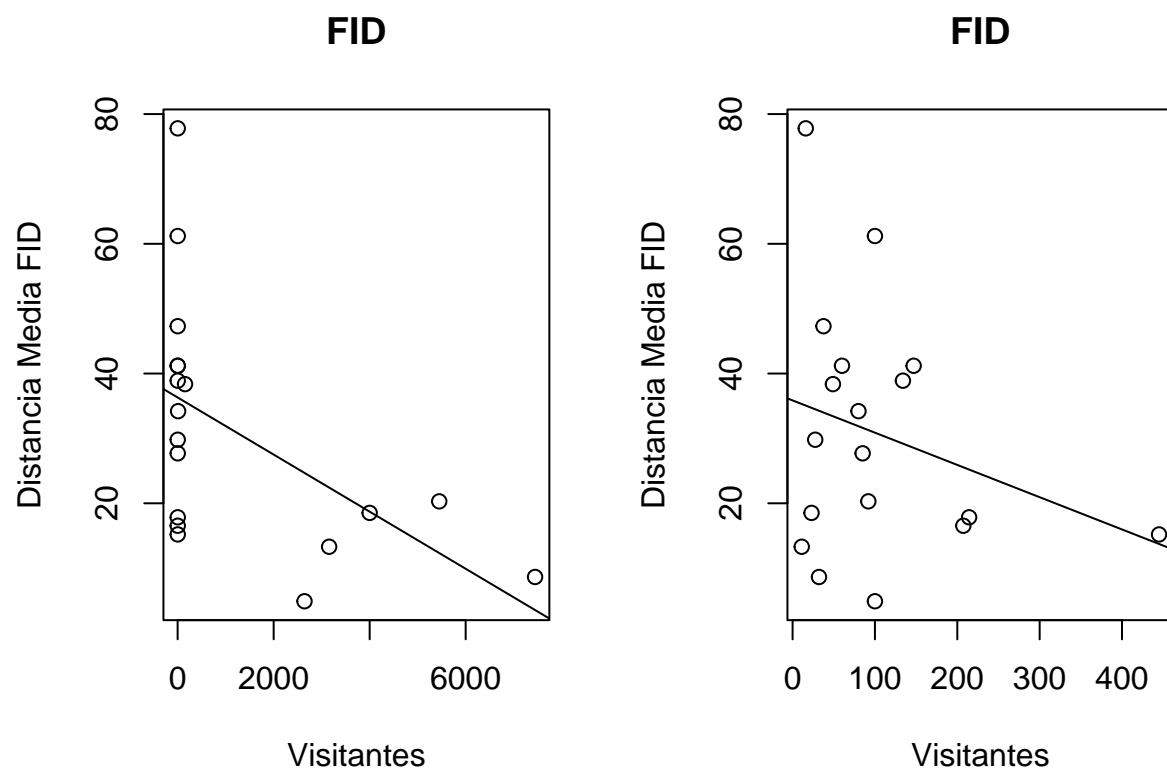
```
ggcorrplot(cor(datos[, 2:4]), hc.order = T, type = "lower", lab = "TRUE")
```



Buscaremos un modelo estadístico que apoye esta idea, pero parece haber cierta relación inversa donde a medida que aumenta el número de visitantes o la densidad de gaviotas tiende a disminuir la distancia a la que inician el vuelo estas.

Esto puede complementarse también con la representación gráfica de *meanFID* frente a *visitors* y *density*.

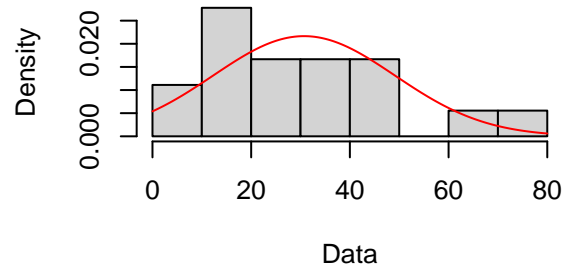
```
par(mfrow = c(1, 2))
plot(datos$visitors, datos$meanFID,
     xlab = "Visitantes", ylab = "Distancia Media FID",
     main = "FID")
abline(lm(datos$meanFID ~ datos$visitors))
plot(datos$density, datos$meanFID,
     xlab = "Visitantes", ylab = "Distancia Media FID",
     main = "FID")
abline(lm(datos$meanFID ~ datos$density))
```



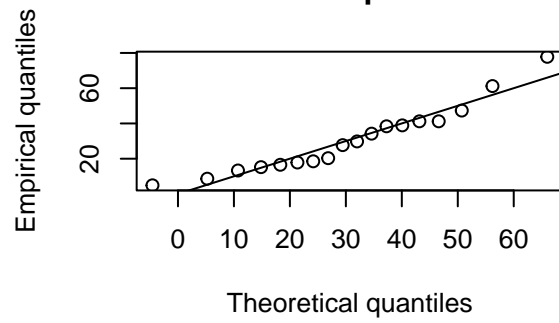
Esta representación gráfica también nos permite cuestionar la idoneidad de un modelo de regresión lineal para representar la realidad que nos ocupa.

```
par(mfrow = c(1,2))
plot(fitdist(datos$meanFID, "norm"))
```

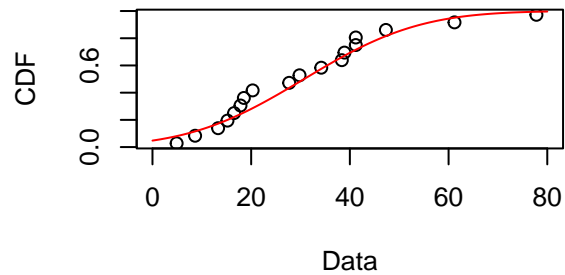
Empirical and theoretical dens.



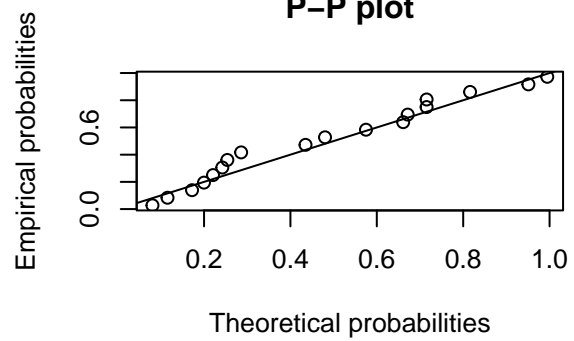
Q-Q plot



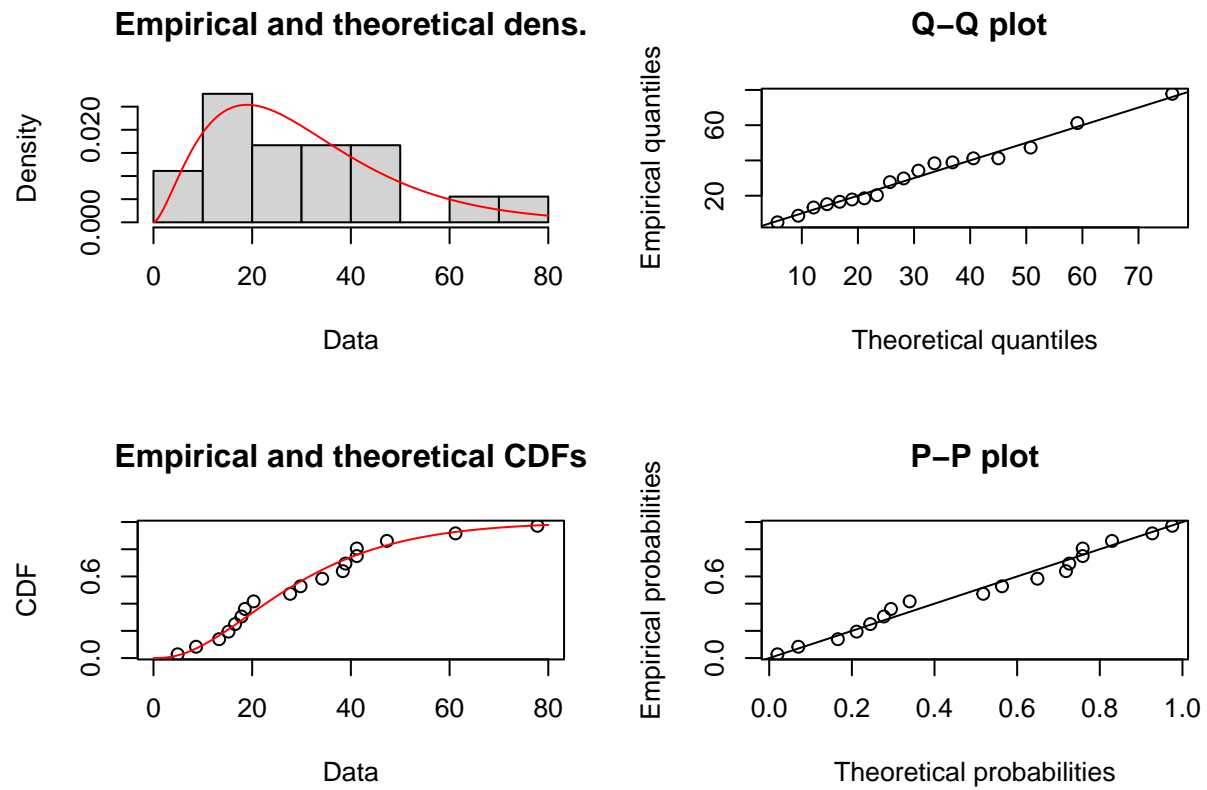
Empirical and theoretical CDFs



P-P plot



```
plot(fitdist(datos$meanFID, "gamma"))
```



Igualmente, a la hora de asumir una distribución para nuestra variable respuesta, resultan igual de razonables la distribución normal y gamma.

MODELIZACIÓN FRECUENTISTA

Hemos realizado inicialmente una serie de modelos básicos que permiten formalizar una primera idea general de lo que sucede en nuestro banco de datos. Asumimos para ello una distribución normal para nuestra variable respuesta. Posteriormente planteamos igualmente estos modelos bajo un modelo lineal generalizado que sigue una distribución gamma.

- Modelo Lineal Normal (1)

```
modelo1 <- lm(meanFID ~ visitors + density, data = datos)
summary(modelo1)

##
## Call:
## lm(formula = meanFID ~ visitors + density, data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.445 -11.363  -1.822   6.743  32.443
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  46.750634   5.767089   8.106 7.31e-07 ***
## visitors     -0.005587   0.001603  -3.486 0.00332 **
## density      -0.086388   0.035033  -2.466 0.02621 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.42 on 15 degrees of freedom
## Multiple R-squared:  0.4892, Adjusted R-squared:  0.4211
## F-statistic: 7.183 on 2 and 15 DF,  p-value: 0.006483
```

Este primer modelo apunta a la significatividad tanto de el número de visitantes como de la densidad de la isla, siendo más relevante la primera covariables.

Una interpretación general apuntaría a que por cada unidad de aumento en el número de visitantes disminuye la distancia de reacción en 0.00559 metros. En otras palabras, por cada 1.000 visitantes más disminuye la distancia de vuelo en aproximadamente 6 metros. La interpretación asociada a la variable density es muy similar.

- Modelo Lineal Normal (2)

Se propone la recategorización de la variable *visitors*

```
datos$visitors2 <- cut(datos$visitors, breaks = 3, labels = c(1, 2, 3), include.lowest = T)
datos$visitors2 <- as.factor(datos$visitors2)

modelo2 <- lm(meanFID ~ visitors2 + density, data = datos)
summary(modelo2)
```

```
##
## Call:
## lm(formula = meanFID ~ visitors2 + density, data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.613  -8.297  -2.127   5.512  30.315
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.97683     5.48117   8.935 3.7e-07 ***
## visitors22  -32.56848     8.94989  -3.639 0.00268 **
## visitors23  -28.72240    10.36165  -2.772 0.01498 *
## density      -0.09324     0.03271  -2.851 0.01283 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.38 on 14 degrees of freedom
## Multiple R-squared:  0.5896, Adjusted R-squared:  0.5016
## F-statistic: 6.704 on 3 and 14 DF,  p-value: 0.004932
```

El detalle más importante del modelo incide en que un nivel medio o alto de visitantes es significativo respecto a un nivel bajo. Puede ser igualmente relevante conocer también si existe una significatividad entre el número medio y alto de visitantes o llega un momento en que quizá tal aumento de la distancia empieza a estabilizarse.

Construimos un nuevo modelo que incide en este aspecto.

```
datos$visitors2 <- relevel(datos$visitors2, ref = 2)
modelo2.1 <- lm(meanFID ~ visitors2 + density, data = datos)
summary(modelo2.1)
```

```
##
## Call:
## lm(formula = meanFID ~ visitors2 + density, data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.613  -8.297  -2.127   5.512  30.315
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  16.40835     7.86455   2.086 0.05572 .
## visitors21  32.56848     8.94989   3.639 0.00268 **
## visitors23   3.84609    12.23127   0.314 0.75782
## density      -0.09324     0.03271  -2.851 0.01283 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.38 on 14 degrees of freedom
## Multiple R-squared:  0.5896, Adjusted R-squared:  0.5016
## F-statistic: 6.704 on 3 and 14 DF,  p-value: 0.004932
```

Interesante ver como la diferencia entre los grupos 2 y 3 no es especialmente significativa.

- Modelo Lineal Mixto

Queremos valorar igualmente si existe un efecto aleatorio en función de la isla de estudio. Analizamos esa variabilidad en función de la isla desde el punto de vista del término independiente.

```
modelo3 <- lme(meanFID ~ visitors + density, random = ~ 1 | colony, data = datos)
summary(modelo3)
```

```
## Linear mixed-effects model fit by REML
##   Data: datos
##       AIC      BIC    logLik
##  165.8726 169.4128 -77.93628
##
## Random effects:
##   Formula: ~1 | colony
##      (Intercept) Residual
## StdDev:      13.50647  5.064927
##
## Fixed effects: meanFID ~ visitors + density
##              Value Std.Error DF   t-value p-value
## (Intercept) 46.75063  5.767089 15   8.106452  0.0000
## visitors    -0.00559  0.001603 15  -3.486020  0.0033
## density     -0.08639  0.035033 15  -2.465897  0.0262
## Correlation:
##      (Intr) vistrs
## visitors -0.542
## density  -0.734  0.300
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -0.4489818 -0.2765843 -0.0443423  0.1641298  0.7897037
##
## Number of Observations: 18
## Number of Groups: 18
```

El hecho de que la varianza asociada a nuestro intercepto sea mayor a la varianza residual ($\sigma_a = 13.51 > 5.06 = \sigma$) apunta a un efecto aleatorio importante en función de la isla de estudio. Los efectos fijos se comportan de manera similar a las construcciones previas

La conclusión que extraemos en este sentido es que si el estudio se ha realizado sobre todas las posibles islas de nuestra región, podríamos plantearnos extraer conclusiones sobre el compartamiento de estas islas dado que existe una variabilidad importante. Sería así pues interesante extraer quizá información sobre las distintas condiciones climatológicas de estas islas. Si realmente solo hemos escogido una serie de islas de un amplio conjunto debemos conocer que existe una amplia variabilidad en los resultados entre las distintas zonas. Conocidas las características de una determinada zona, podríamos inferir el comportamiento en islas cercanas no estudiadas en nuestro análisis.

- Modelo Aditivo

Las dos gráficas del análisis descriptivo invitaban a una relación no lineal de nuestras distintas covariables con la variable respuesta. Planteamos la construcción de un modelo suavizado.


```

modelo4 <- gam(meanFID ~ s(visitors, bs = "cp") + density,
               family = gaussian,
               data = datos)
summary(modelo4)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## meanFID ~ s(visitors, bs = "cp") + density
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.99789    5.44725   6.976 4.67e-06 ***
## density    -0.07042    0.03806  -1.850  0.0843 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(visitors) 1.124     6 0.935  0.0221 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.29   Deviance explained = 37.8%
## GCV = 308.96   Scale est. = 255.34      n = 18

```

El reducido número de observaciones nos ha dificultado mucho la elección de una base para nuestro modelo al tiempo que imposibilita la suavización de las dos covariables. Hemos decidido en este sentido suavizar la variable *visitors* siendo esta suavización significativa.

- Resumen Resultados

Así quedan los distintos AIC para los distintos modelos planteados.

```

kable(data.frame(Modelo = c("Modelo 1", "Modelo 2", "Modelo 3", "Modelo 4"),
                  AIC = c(AIC(modelo1), AIC(modelo2), AIC(modelo3), AIC(modelo4))))

```

Modelo	AIC
Modelo 1	151.8825
Modelo 2	149.9450
Modelo 3	165.8726
Modelo 4	155.6651

Se observa que la inclusión del efecto aleatorio o la suavización de alguna de nuestras variables no supone ninguna mejora con respecto a nuestros dos modelos iniciales más simples. De esta manera, escogeríamos desde un enfoque frecuentista uno de los dos primeros modelos. Creemos que la elección entre el primer o el segundo modelo debería responder a las pretensiones de nuestro estudio.

Hemos decidido así replantear los dos primeros modelos pero asumiendo una distribución gamma para nuestra variable respuesta.

- Modelo Lineal Generalizado: Gamma (1)

```
modelo5 <- glm(meanFID ~ visitors + density, data = datos, family = Gamma(link = "log"))
modelo5$aic
```

```
## [1] 145.4816
```

El AIC de este modelo supone una mejora considerable con respecto a los modelos anteriores bajo distribución normal. En este sentido, seguimos planteando modelos, estudiando por una parte trabajar con nuestras variables explicativas de manera polinómica o viendo que tipo de link pudiera ofrecer mejores resultados.

```
AICs_ModeloInverse <- c()
Deviances_ModeloInverse <- c()
for(i in 1:7){
  modelo5INV <- glm(meanFID ~ poly(visitors, i) + poly(density, i), data = datos, family = Gamma(link = "log"))
  AICs_ModeloInverse[i] <- modelo5INV$aic
  Deviances_ModeloInverse[i] <- (modelo5INV$null.deviance - modelo5INV$deviance)/modelo5INV$null.deviance
}

AICs_ModeloLog <- c()
Deviances_ModeloLog <- c()
for(i in 1:7){
  modelo5LOG <- glm(meanFID ~ poly(visitors, i) + poly(density, i), data = datos, family = Gamma(link = "log"))
  AICs_ModeloLog[i] <- modelo5LOG$aic
  Deviances_ModeloLog[i] <- (modelo5LOG$null.deviance - modelo5LOG$deviance)/modelo5LOG$null.deviance
}
```

```
kable(data.frame(Orden = c(1:7),
  AIC_Inverse = AICs_ModeloInverse,
  AIC_Log = AICs_ModeloLog,
  Deviance_Inverse = Deviances_ModeloInverse,
  Deviance_Log = Deviances_ModeloLog))
```

Orden	AIC_Inverse	AIC_Log	Deviance_Inverse	Deviance_Log
1	142.6936	145.4816	60.32809	53.88320
2	145.5089	147.5861	62.79328	58.36666
3	140.3829	134.6332	77.36842	83.48845
4	137.8276	130.1772	84.20348	89.63487
5	133.2633	128.3155	90.14463	92.50126
6	131.5464	132.1812	92.81338	92.55671
7	132.8148	134.1059	93.82106	93.36365

En este momento, debemos analizar si priorizar un buen AIC y una buena Deviance explicada a pesar de quizá cierta pérdida de interpretabilidad o utilizar un modelo más simple, con mayor margen de error pero más fácil de entender.

Desde nuestro punto de vista, la mejora en la capacidad de ajuste que supone el trabajo con nuestras covariables de forma polinómica no compensa la capacidad de interpretación que perdemos a la hora de intentar explicar a alguien como influyen la densidad de gaviotas y el número de visitantes en la distancia de reacción.

Por esta manera decidimos no incluir una regresión polinómica en nuestro estudio, aunque merece la pena comentar la mejora en los resultados que esta produce. Decidimos seguir trabajando igualmente con el link *log* por ser más sencillo de interpretar.

- Modelo Lineal Generalizado: Gamma (2)

```
datos$visitors2 <- cut(datos$visitors, breaks = 3, labels = c(1, 2, 3), include.lowest = T)
datos$visitors2 <- as.factor(datos$visitors2)
modelo6 <- glm(meanFID ~ visitors2 + density, data = datos, family = Gamma(link = "log"))
modelo6$aic
```

```
## [1] 141.2043
```

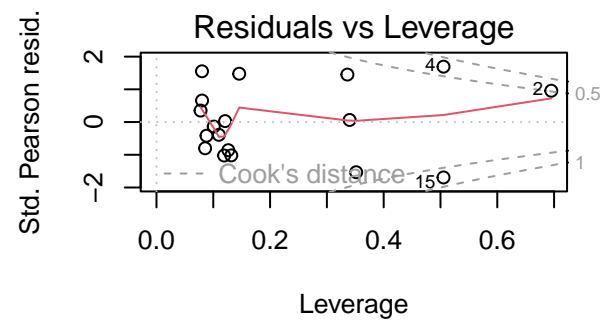
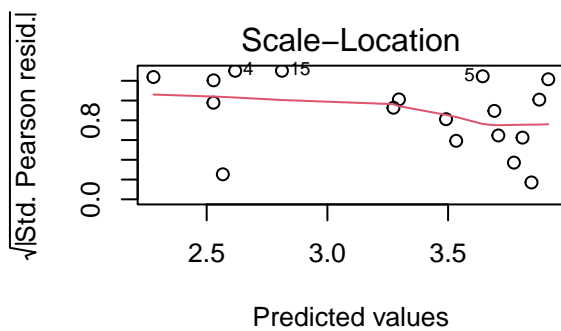
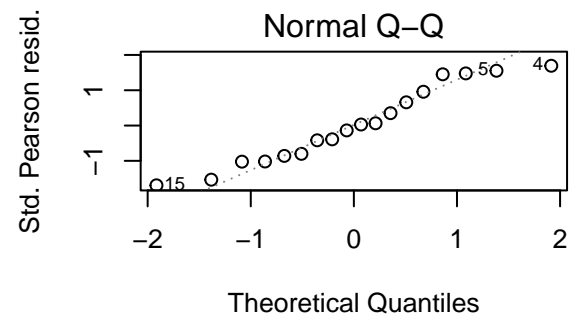
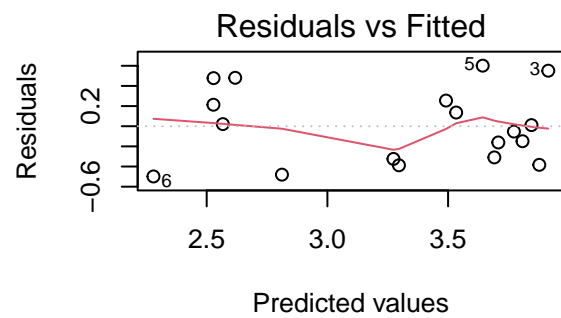
Presentamos a continuación una comparación entre los modelos de regresión normal y gamma que trabajan con la variable *visitors* sin categorizar (*modelo1* y *modelo5*) y los modelos que plantean su recategorización (*modelo2* y *modelo6*)

```
kable(data.frame(Modelo = c("Modelo 1 (Normal sin Categorizar)",
                           "Modelo 2 (Normal Categorizada)",
                           "Modelo 5 (Gamma sin Categorizar)",
                           "Modelo 6 (Gamma Categorizada)"),
               AIC = c(AIC(modelo1), AIC(modelo2), AIC(modelo5), AIC(modelo6)),
               R2_Deviance = c(summary(modelo1)$r.squared,
                               summary(modelo2)$r.squared,
                               (modelo5$null.deviance - modelo5$deviance)/modelo5$null.deviance*100,
                               (modelo6$null.deviance - modelo6$deviance)/modelo6$null.deviance*100))
```

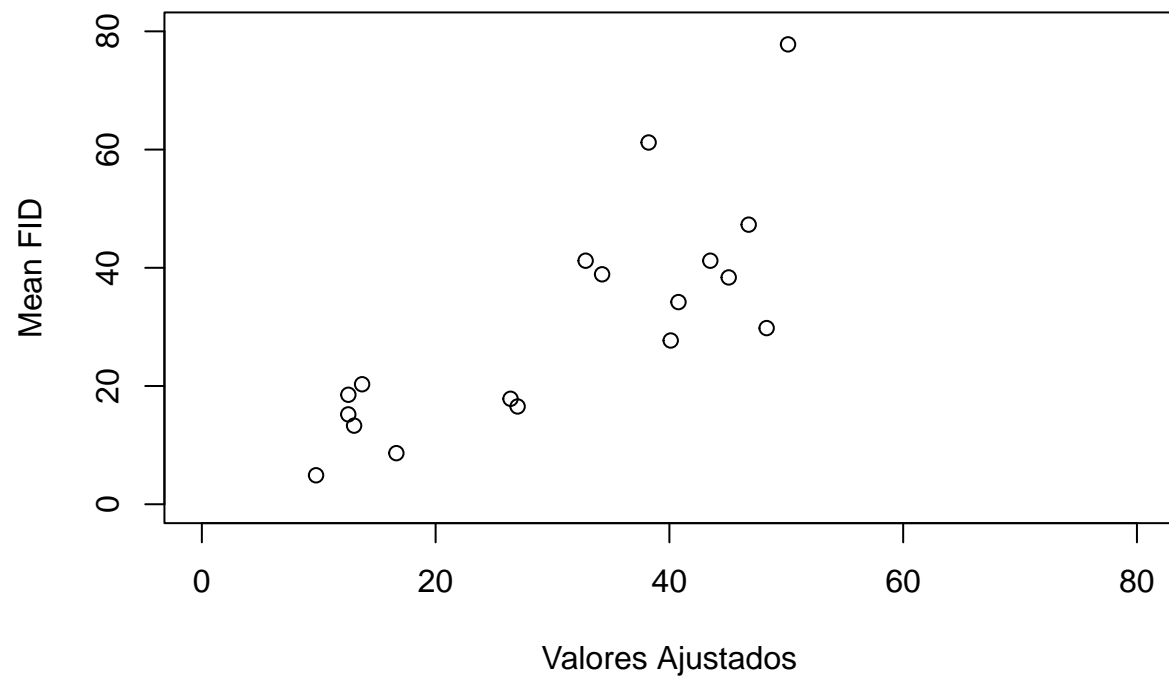
Modelo	AIC	R2_Deviance
Modelo 1 (Normal sin Categorizar)	151.8825	0.4892143
Modelo 2 (Normal Categorizada)	149.9450	0.5895702
Modelo 5 (Gamma sin Categorizar)	145.4816	53.8831954
Modelo 6 (Gamma Categorizada)	141.2043	67.1676081

Por todo ello, consideramos que el último modelo, es el que nos puede aportar una mayor información en torno al comportamiento de las gaviotas en presencia de humanos. Planteamos a continuación la construcción de estos modelos desde el punto de vista Bayesiano, mostrando antes como nuestro mejor modelo (*modelo6*) verifica las condiciones de aplicabilidad, y como queda el ajuste que plantea.

```
par(mfrow=c(2, 2))
plot(modelo6)
```



```
par(mfrow=c(1, 1))
plot(x = modelo6$fitted.values, y = datos$meanFID,
     xlab = "Valores Ajustados", ylab = "Mean FID",
     xlim = c(0, 80), ylim = c(0, 80))
```



MODELIZACIÓN BAYESIANA

Se ha realizado la construcción bayesiana desde dos herramientas distintas. WinBugs e INLA, se presentan los resultados para ambas y se comparan estos con el mejor modelo frecuentista propuesto.

- WinBugs

```
set.seed(123)
bayesiano <- function()
{
  # Verosimilitud
  for (i in 1:N){
    meanFID[i] ~ dgamma(r, shape[i])
    shape[i] <- r/mu[i]
    log(mu[i]) <- alpha + beta.v[visitors[i]] + beta.d*density[i]
  }

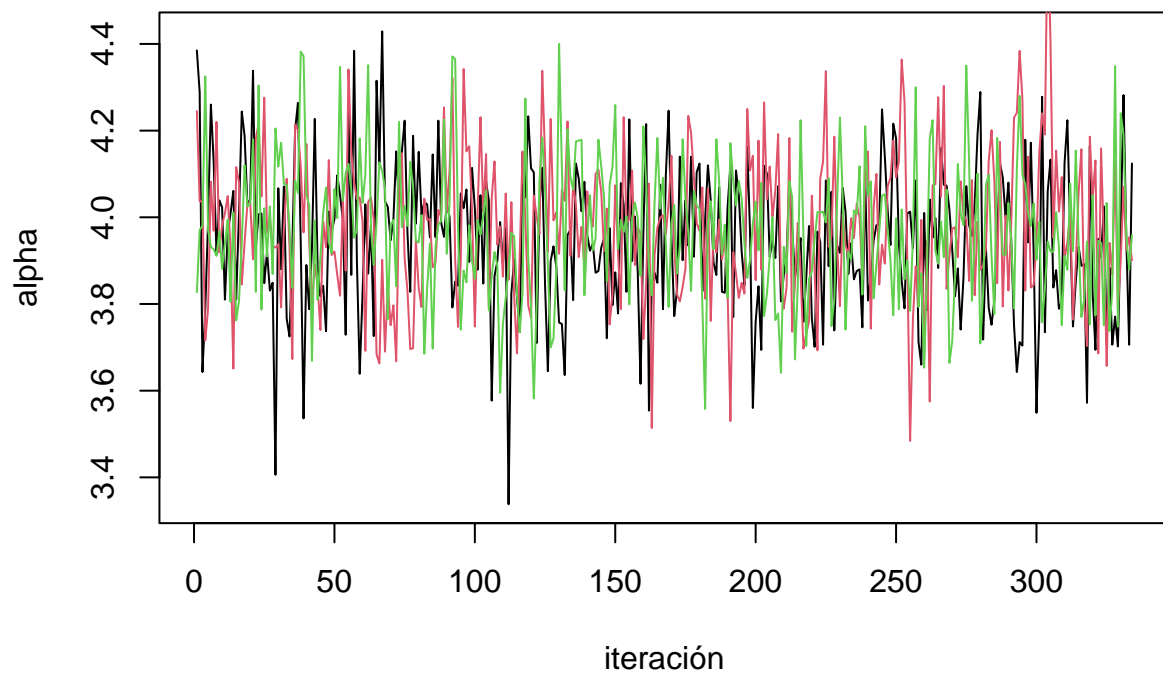
  # Distribuciones previas
  alpha ~ dnorm(0, 0.00001)
  beta.d ~ dnorm(0, 0.00001)
  # Corner constrain
  beta.v[1] <- 0
  for (j in 2:3){
    beta.v[j] ~ dnorm(0, 0.00001)
  }
  r ~ dunif(0, 100)
}

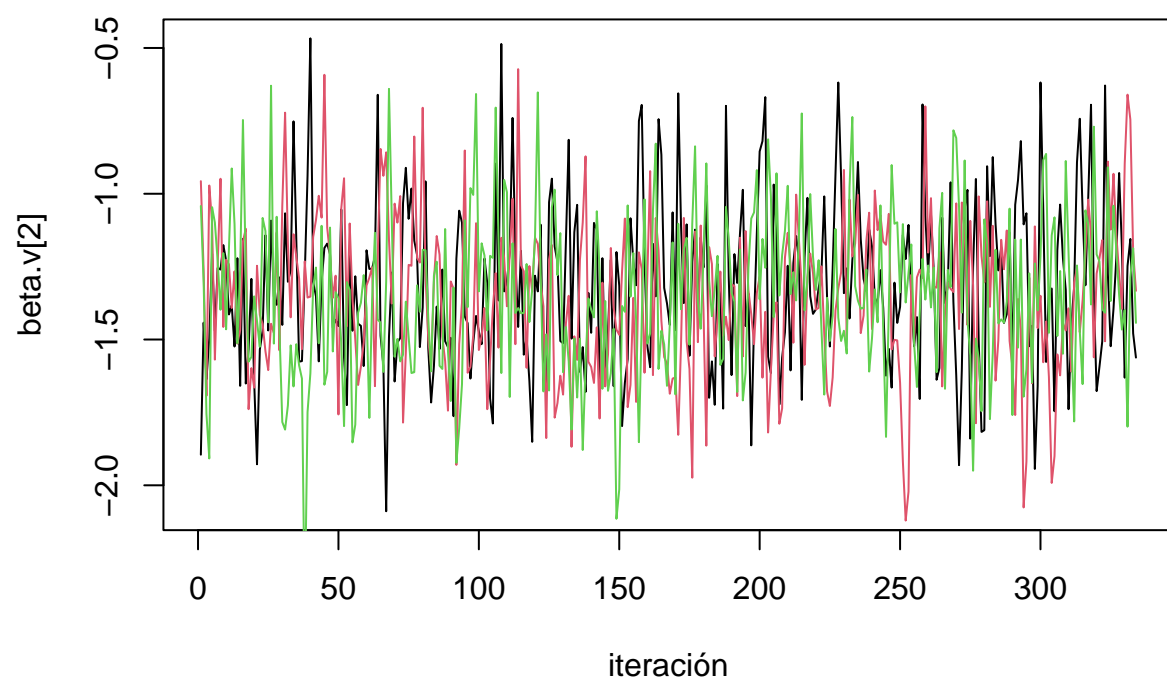
datos1 <- list(N = dim(datos)[1],
              meanFID = datos$meanFID,
              visitors = as.numeric(datos$visitors2),
              density = datos$density)

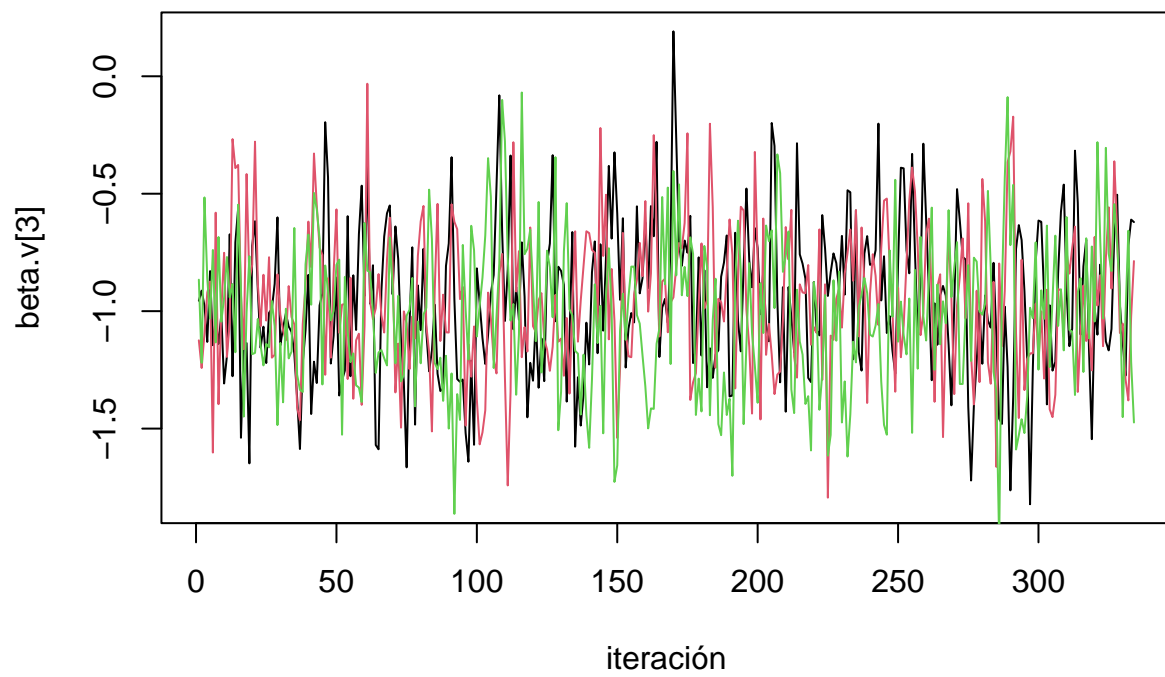
iniciales <- function()
{
  list(alpha = rnorm(1), beta.v = c(NA, rnorm(1), rnorm(1)), beta.d = rnorm(1), r = runif(1))
}

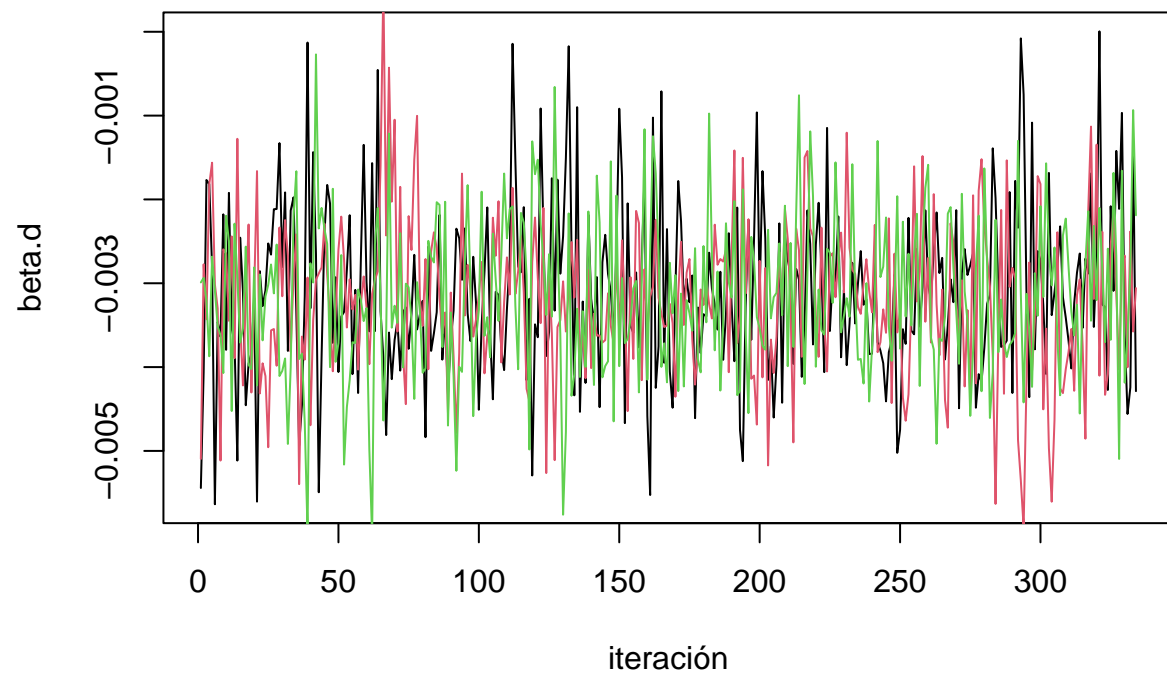
parametros <- c("alpha", "beta.v", "beta.d", "mu", "r")
iteraciones <- 100000
burnin <- 10000
Resulbayesiano <- bugs(model = bayesiano, data = datos1, inits = iniciales,
                      param = parametros, n.iter = iteraciones, n.burnin = burnin)

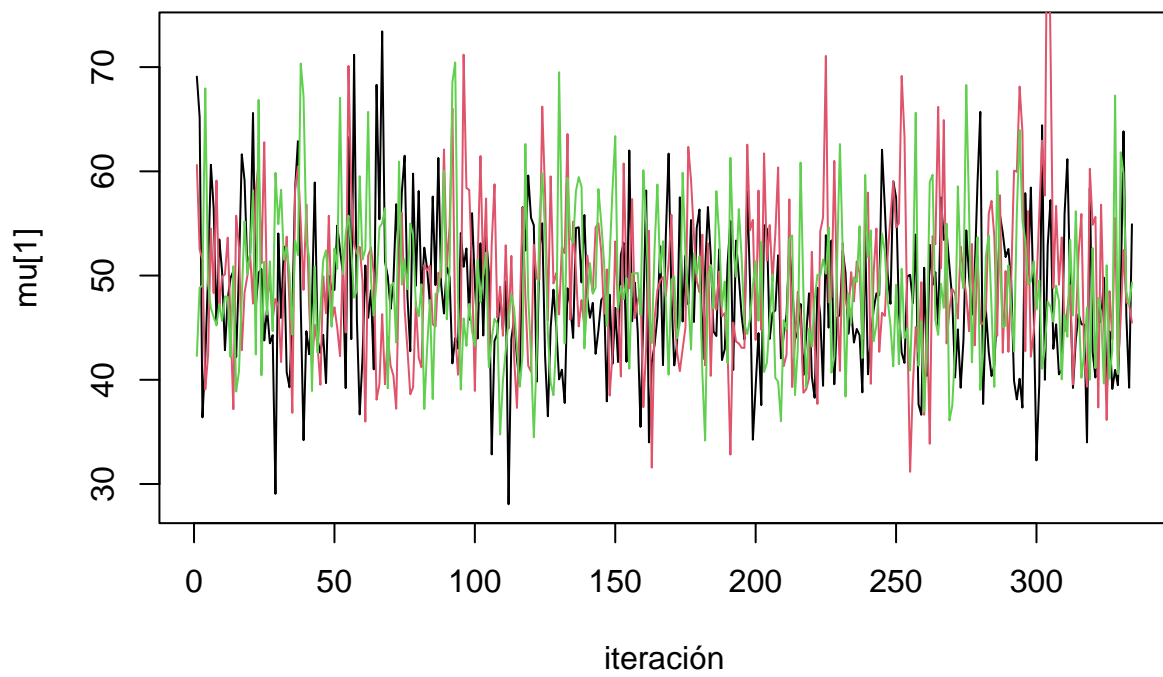
ResultadoBayesiano <- Resulbayesiano$sims.list
for (i in 1:24){
  plot(Resulbayesiano$sims.array[, 1, i], type = "l",
       xlab = "iteración", ylab = dimnames(Resulbayesiano$sims.array)[[3]][i])
  lines(Resulbayesiano$sims.array[, 2, i], col = 2)
  lines(Resulbayesiano$sims.array[, 3, i], col = 3)
}
```

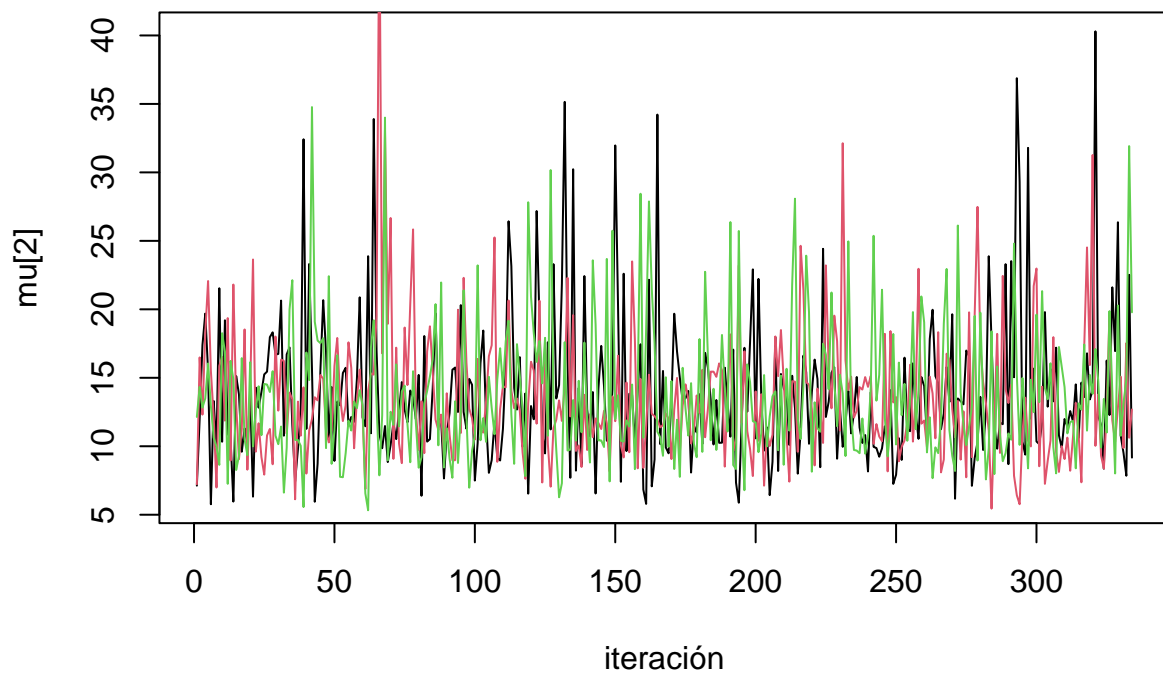


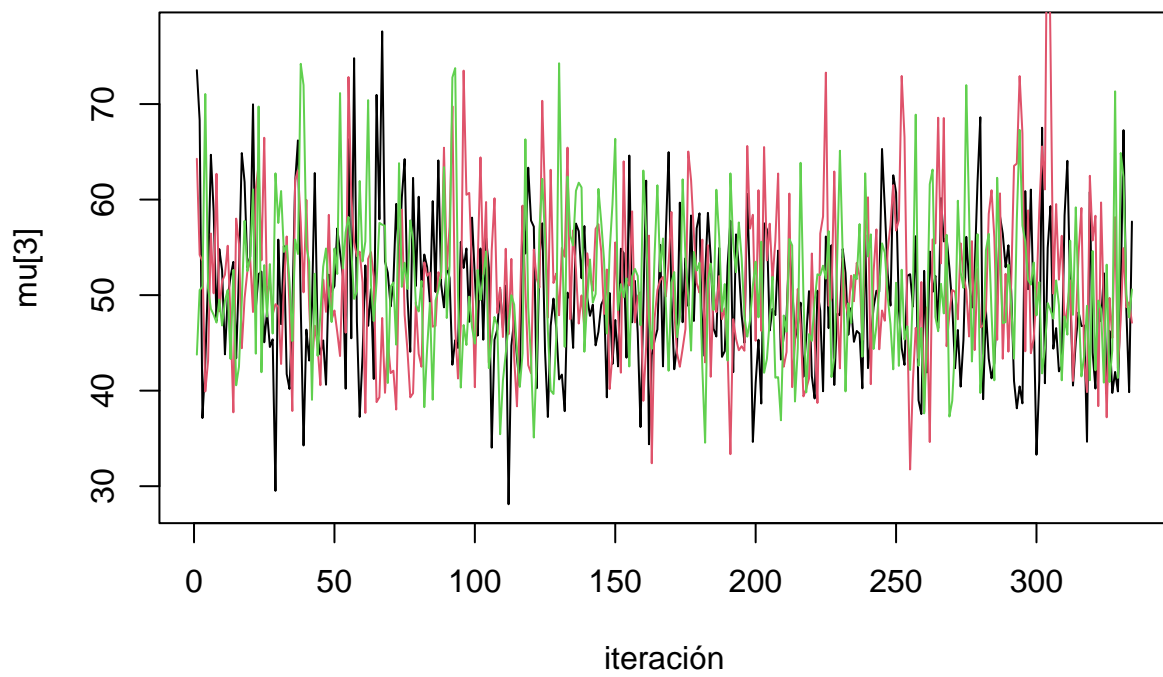


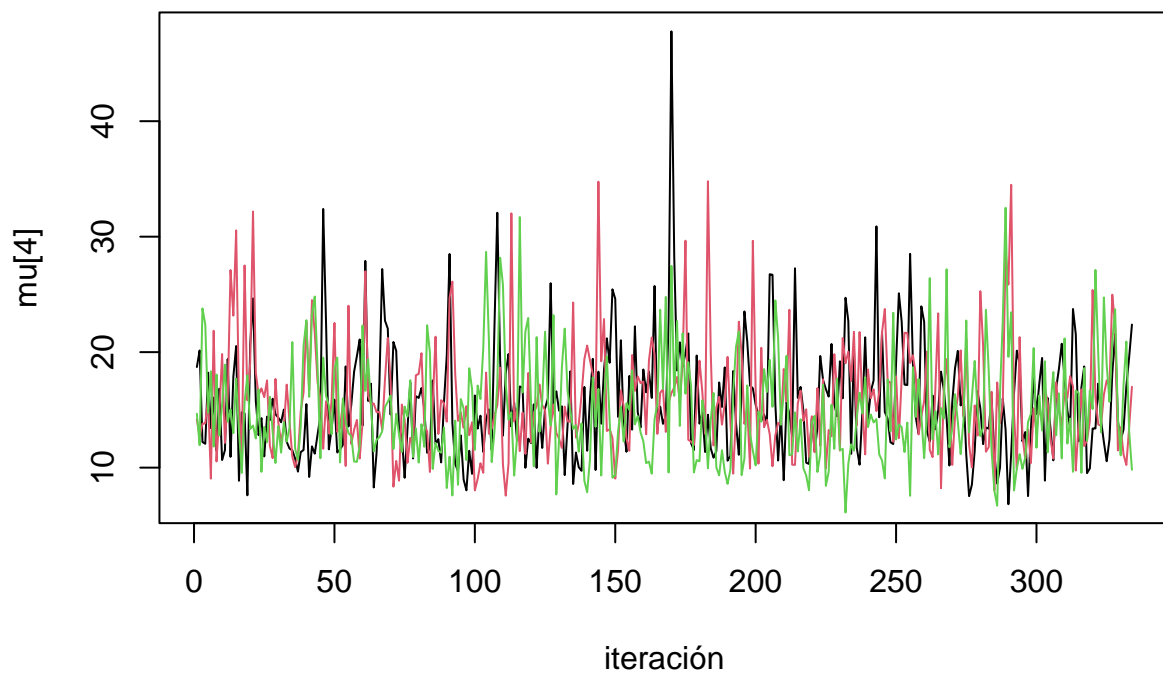


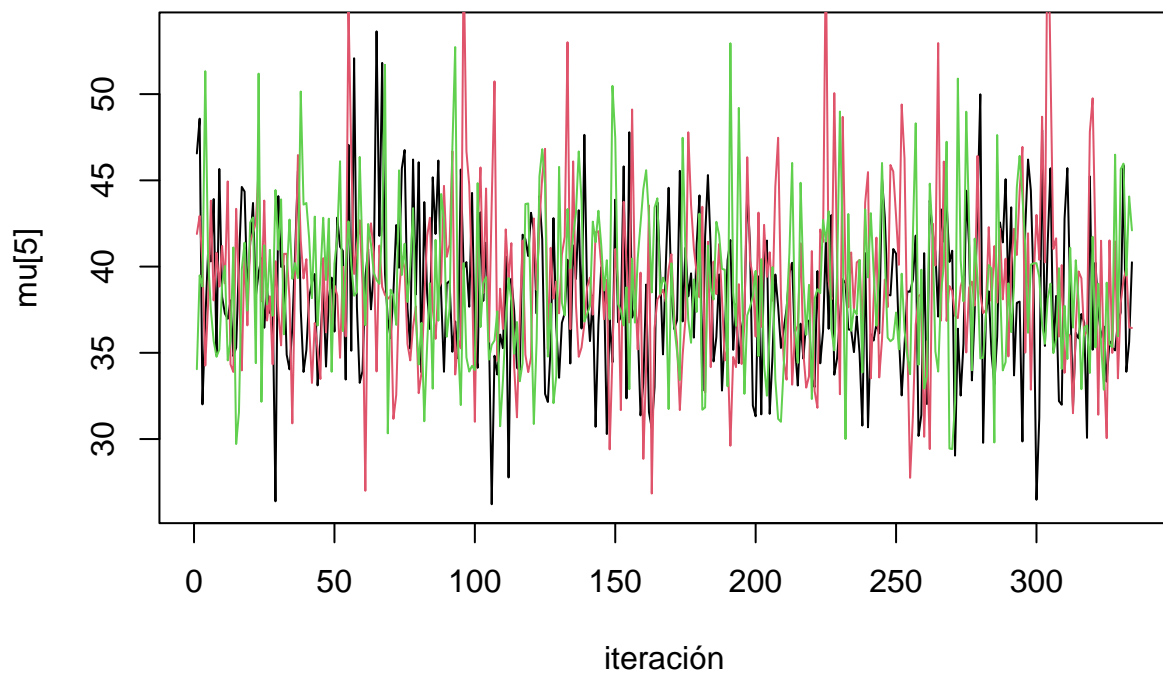


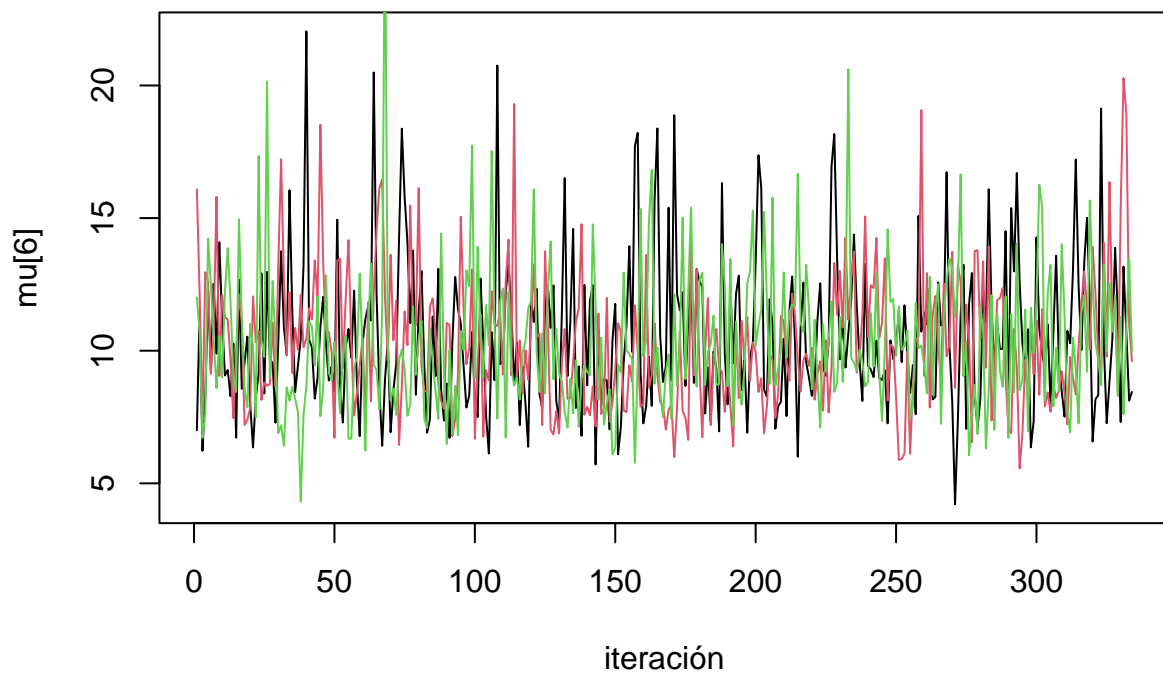


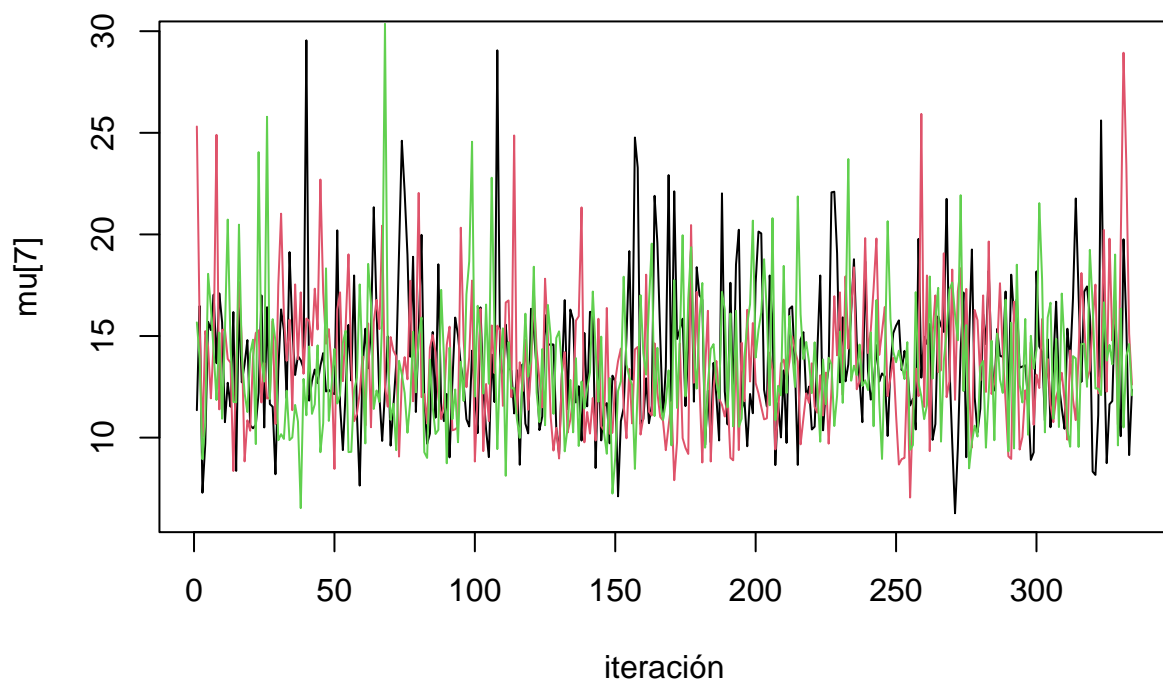


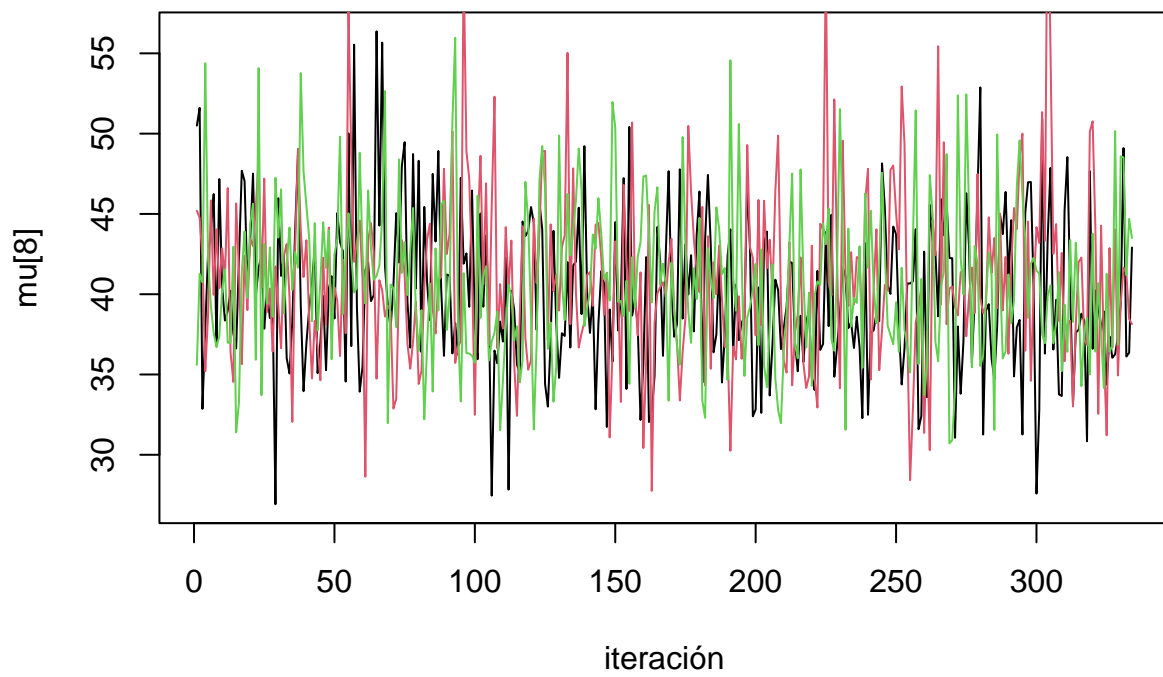


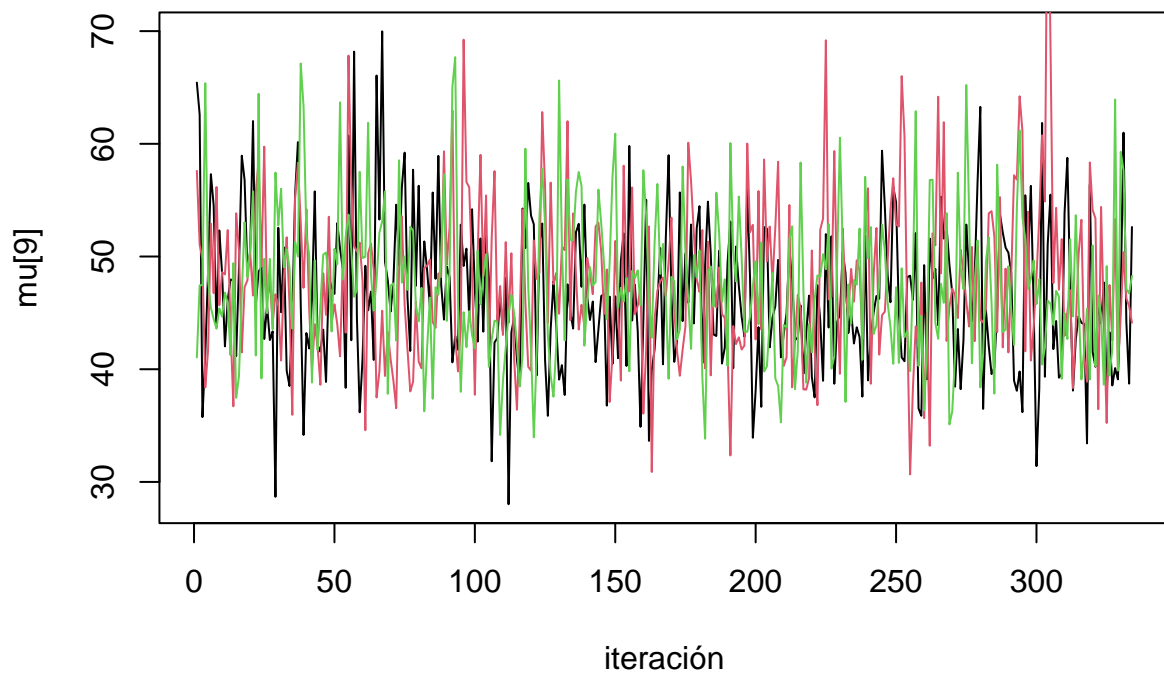


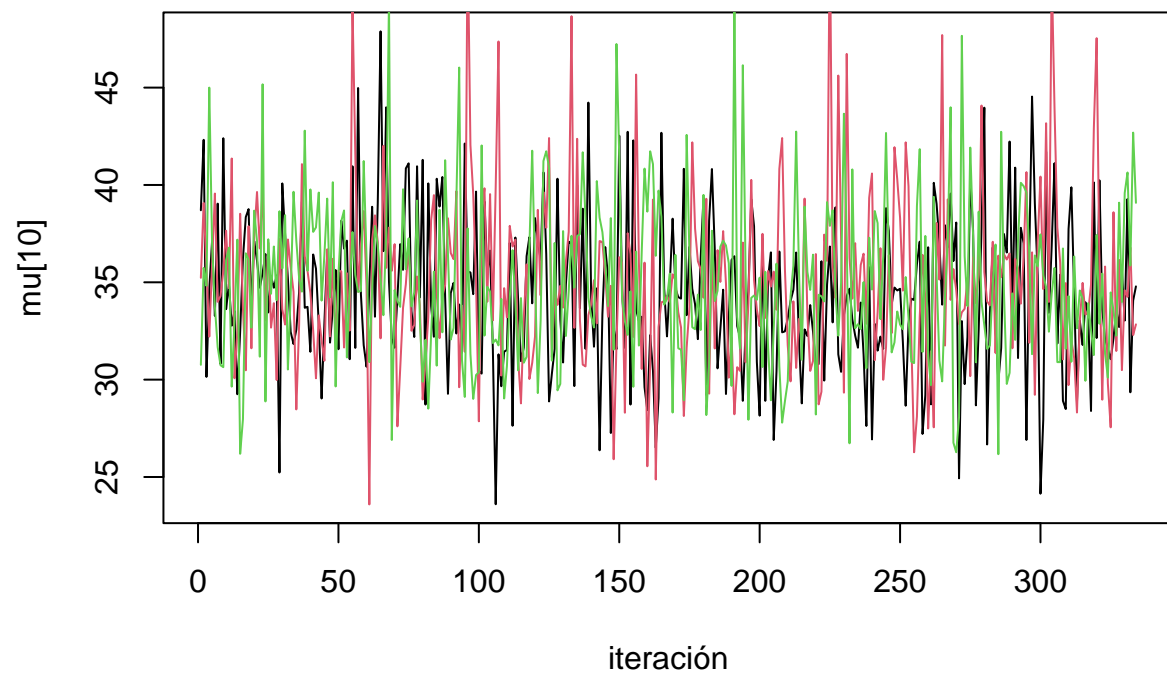


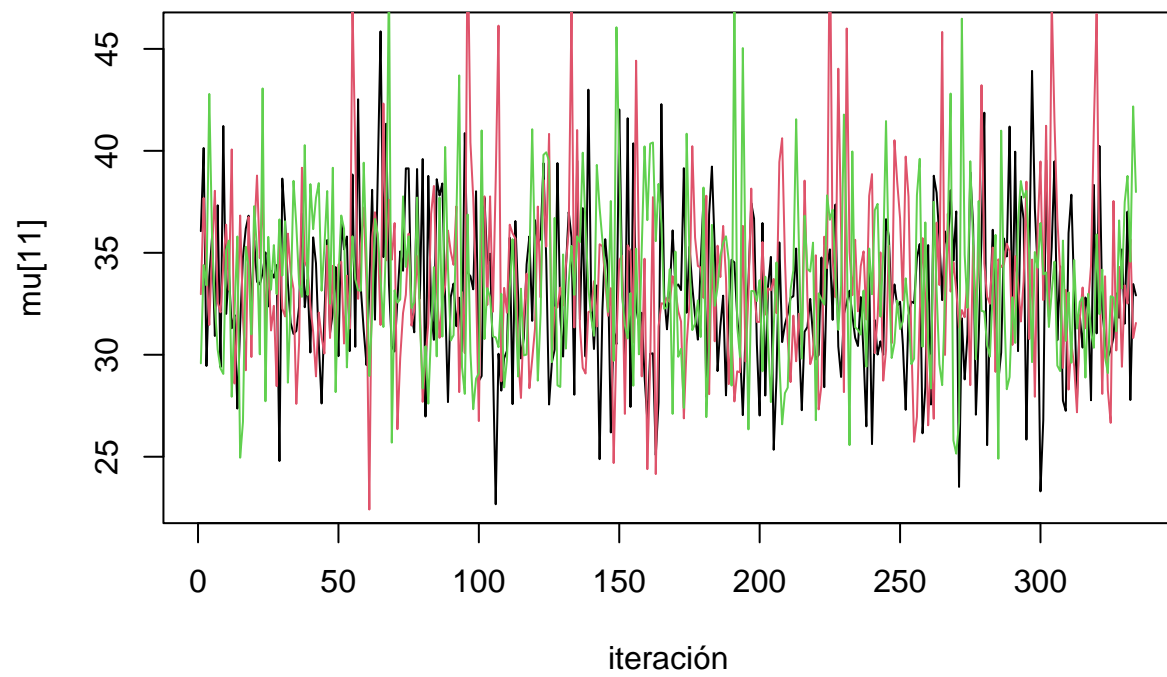


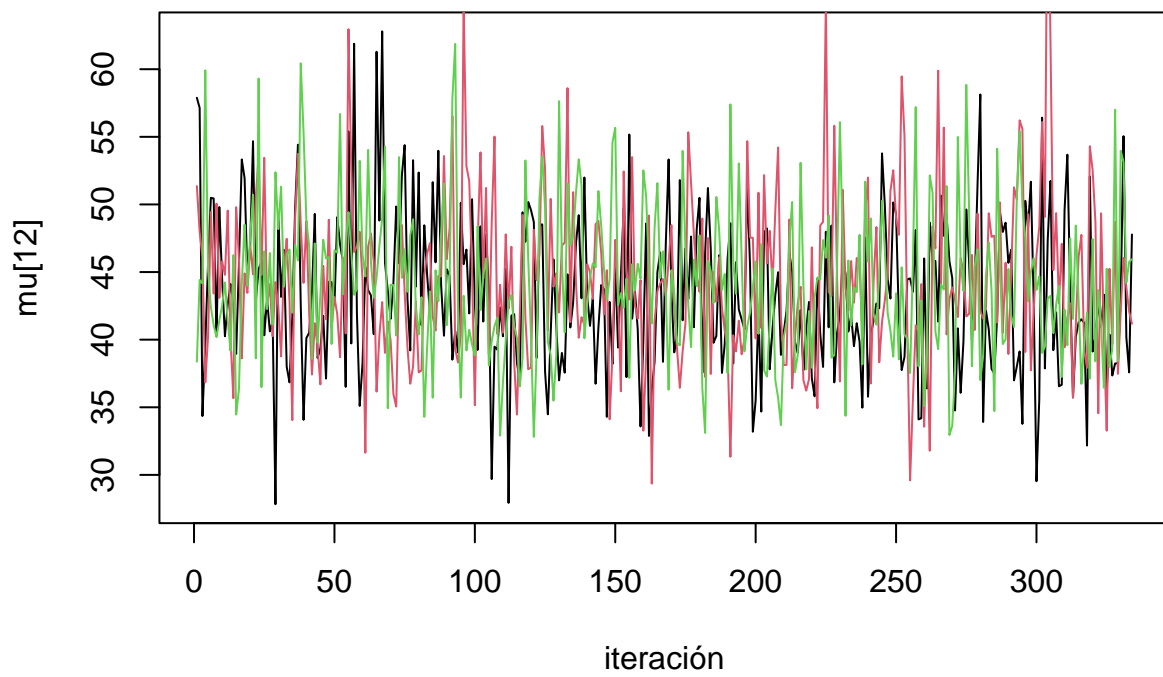


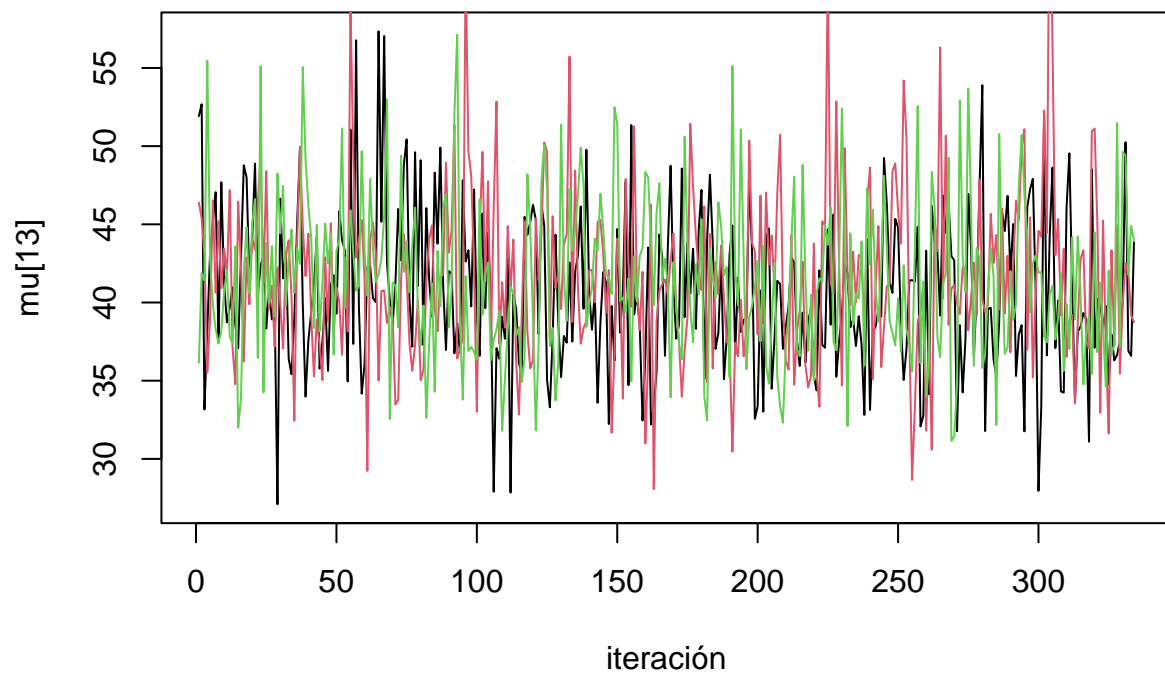


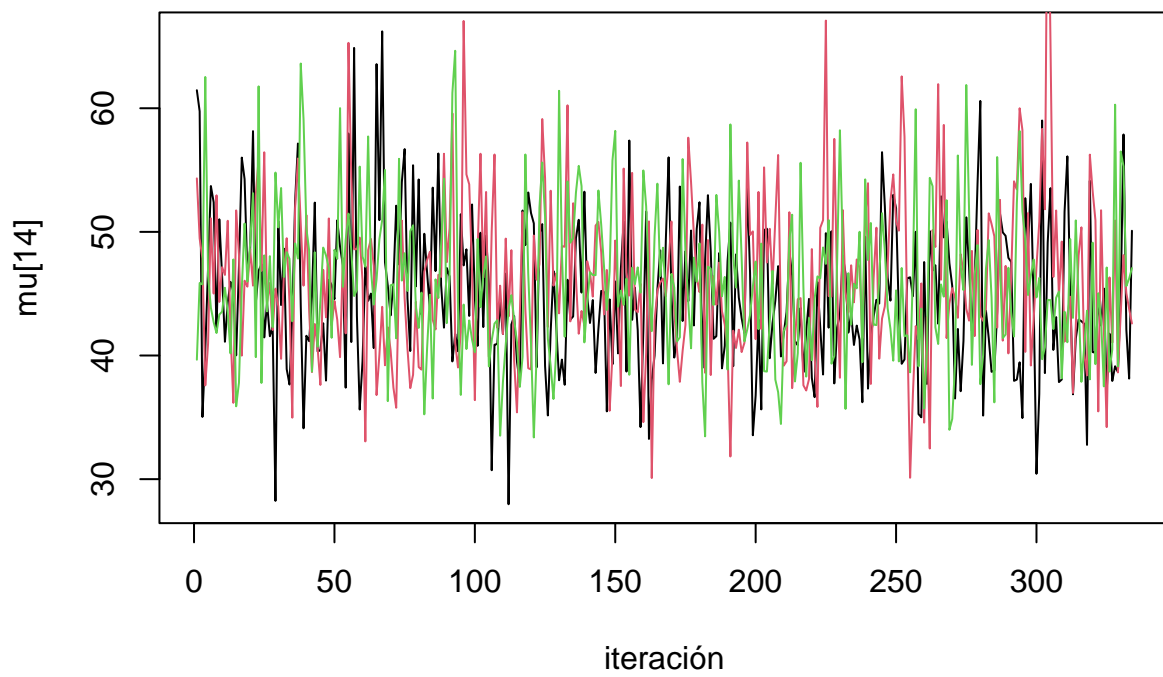


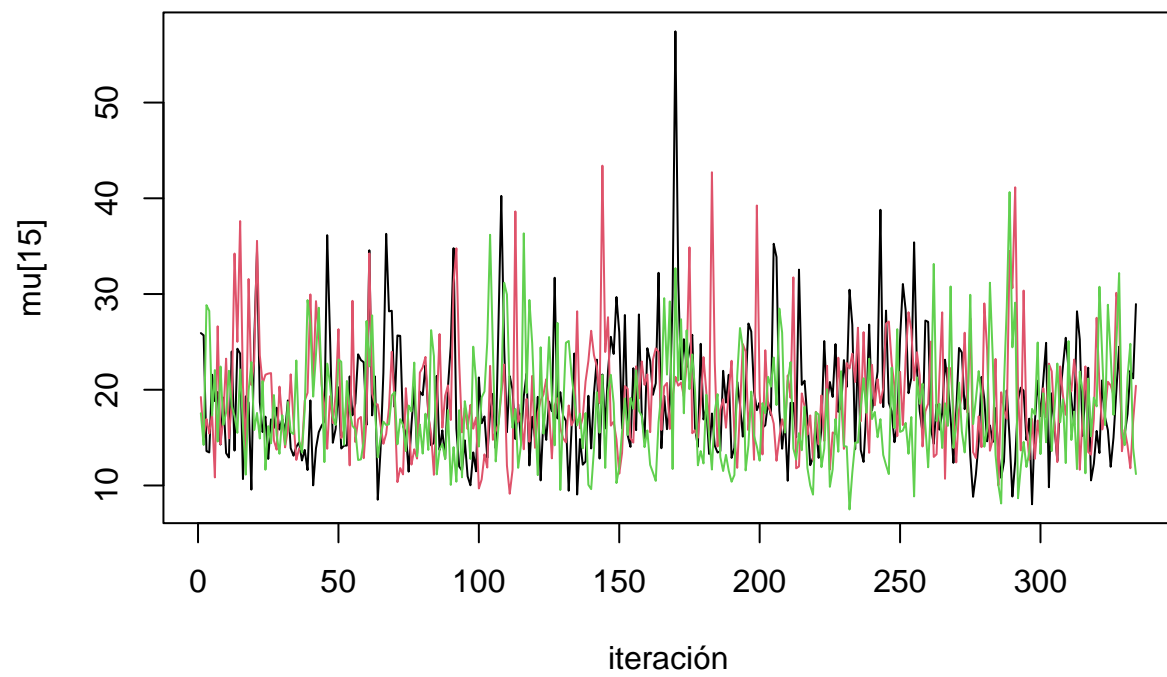


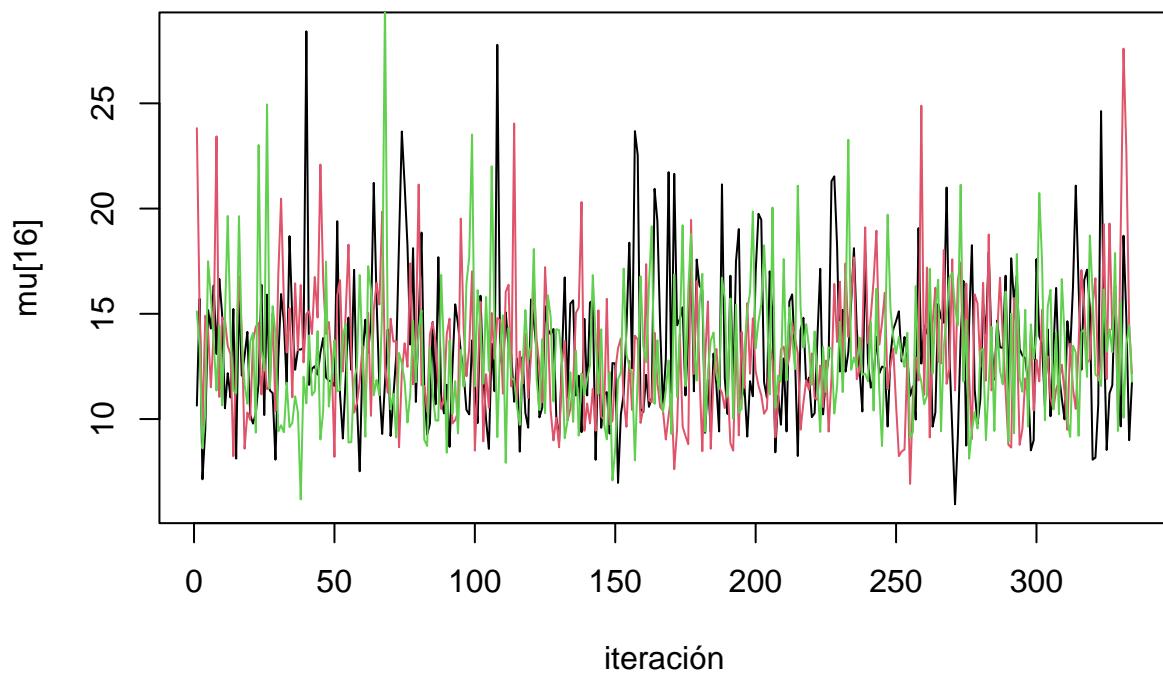


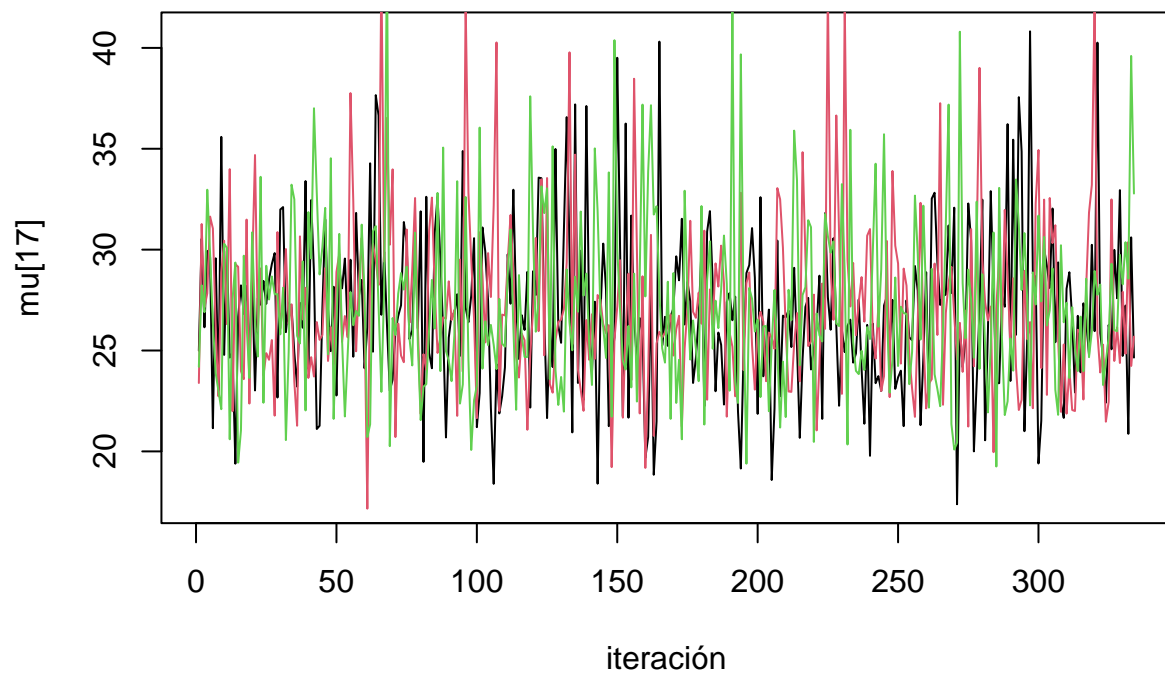


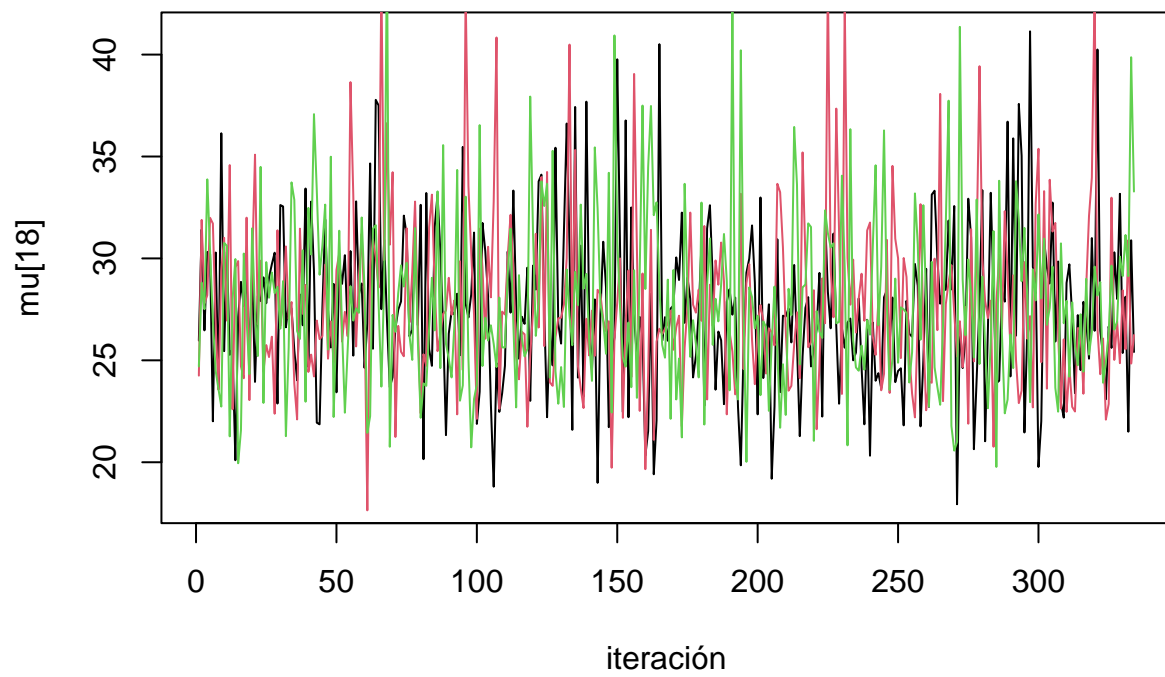


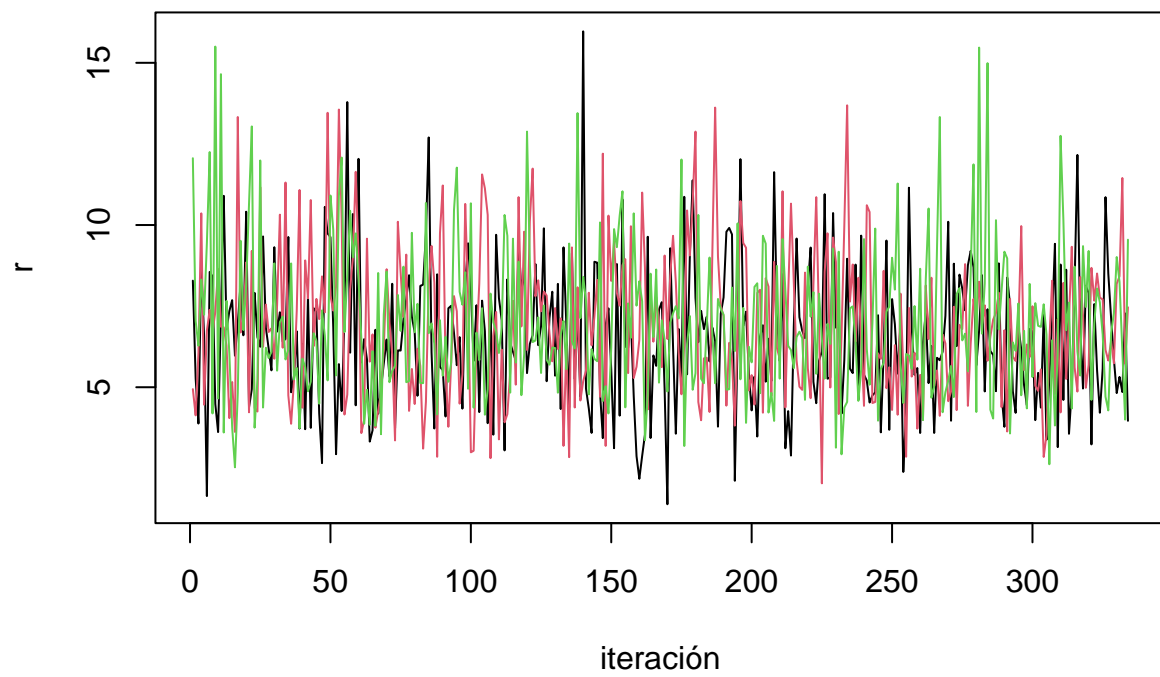


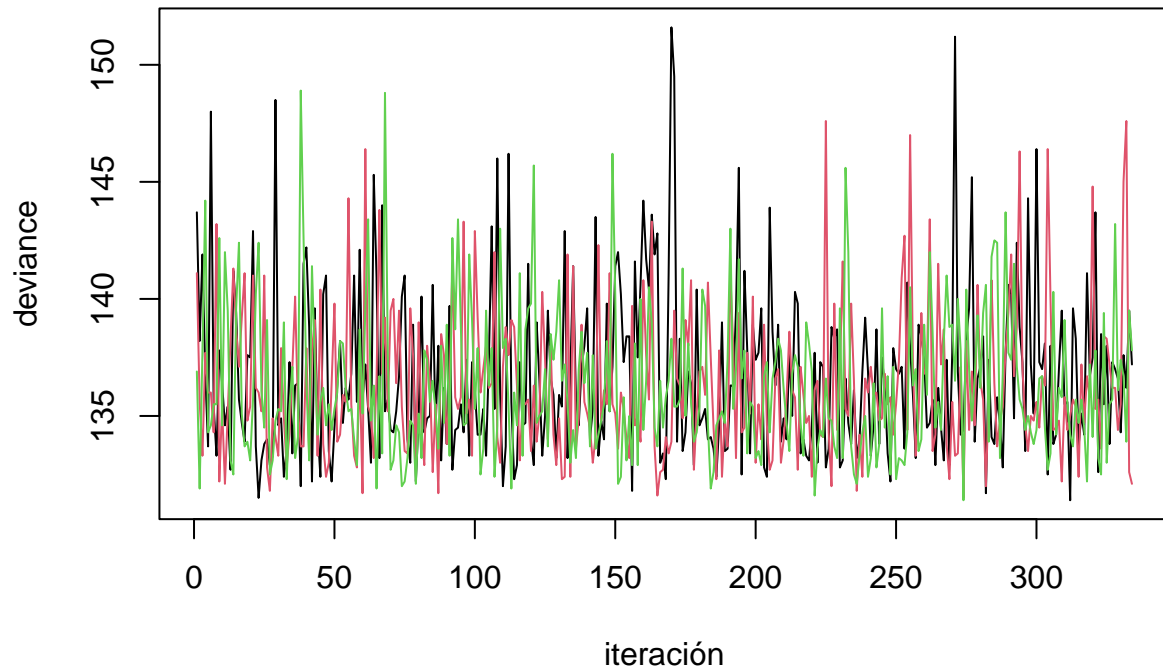












```
round(Resulbayesiano$summary, 3)
```

##	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
## alpha	3.965	0.164	3.660	3.857	3.964	4.069	4.303	1.008	290
## beta.v[2]	-1.340	0.273	-1.852	-1.520	-1.348	-1.164	-0.740	1.004	450
## beta.v[3]	-0.978	0.315	-1.566	-1.192	-0.986	-0.770	-0.317	1.009	210
## beta.d	-0.003	0.001	-0.005	-0.004	-0.003	-0.003	-0.001	1.002	1000
## mu[1]	48.885	7.269	36.833	43.920	48.320	53.000	65.698	1.007	280
## mu[2]	13.732	5.079	7.050	10.340	12.710	15.737	26.428	1.000	1000
## mu[3]	50.743	7.905	37.731	45.335	50.170	55.200	68.883	1.007	290
## mu[4]	15.485	4.732	8.591	12.200	14.715	17.695	27.169	1.009	220
## mu[5]	38.746	4.654	30.621	35.640	38.425	41.357	48.990	1.006	310
## mu[6]	10.385	2.670	6.446	8.525	10.015	11.767	16.818	1.001	1000
## mu[7]	13.714	3.372	8.823	11.482	13.310	15.500	22.018	1.001	1000
## mu[8]	40.628	5.005	31.736	37.142	40.315	43.450	51.447	1.007	290
## mu[9]	47.329	6.768	36.063	42.702	46.785	51.232	62.814	1.007	280
## mu[10]	34.803	4.171	27.551	32.050	34.485	37.087	43.989	1.004	400
## mu[11]	33.413	4.085	26.491	30.685	33.010	35.700	42.319	1.004	470
## mu[12]	44.021	5.808	34.071	40.160	43.575	47.420	56.527	1.007	280
## mu[13]	41.292	5.146	32.210	37.700	40.930	44.257	52.416	1.007	280
## mu[14]	45.606	6.248	35.001	41.382	45.140	49.322	59.578	1.007	280
## mu[15]	18.739	5.828	10.252	14.545	17.765	21.587	33.842	1.009	220
## mu[16]	13.208	3.243	8.500	11.050	12.810	14.827	21.140	1.001	1000
## mu[17]	27.124	4.162	20.550	24.427	26.530	29.310	37.190	1.001	1000
## mu[18]	27.715	4.130	21.051	25.030	27.180	29.932	37.578	1.001	1000

```
## r          6.815 2.267   3.136   5.170   6.607   8.179 12.019 1.009   230
## deviance 136.432 3.264 132.102 134.025 135.700 138.000 144.297 1.006   610
```

Adecuada convergencia de las cadenas y adecuada construcción del modelo.

- INLA

```
Resulinla <- inla(meanFID ~ 1 + visitors2 + density,
  family = 'gamma',
  data = datos,
  control.inla = list(strategy = 'simplified.laplace'))
summary(Resulinla)
```

```
##
## Call:
## c("inla.core(formula = formula, family = family, contrasts = contrasts,
## ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
## scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
## ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
## verbose, ", " lincomb = lincomb, selection = selection, control.compute
## = control.compute, ", " control.predictor = control.predictor,
## control.family = control.family, ", " control.inla = control.inla,
## control.fixed = control.fixed, ", " control.mode = control.mode,
## control.expert = control.expert, ", " control.hazard = control.hazard,
## control.lincomb = control.lincomb, ", " control.update =
## control.update, control.lp.scale = control.lp.scale, ", "
## control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
## ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
## num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
## working.directory = working.directory, ", " silent = silent, inla.mode
## = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
## .parent.frame)")
## Time used:
## Pre = 3.16, Running = 0.247, Post = 0.0186, Total = 3.43
## Fixed effects:
##      mean      sd 0.025quant 0.5quant 0.975quant mode kld
## (Intercept)  3.961 0.164      3.642   3.959      4.293   NA   0
## visitors22  -1.358 0.276     -1.886  -1.365     -0.789   NA   0
## visitors23  -1.051 0.314     -1.640  -1.063     -0.394   NA   0
## density     -0.003 0.001     -0.005  -0.003     -0.001   NA   0
##
## Model hyperparameters:
##                                     mean      sd 0.025quant 0.5quant
## Precision parameter for the Gamma observations 6.83 2.38      3.08      6.53
##                                     0.975quant mode
## Precision parameter for the Gamma observations 11.91  NA
##
## Marginal log-Likelihood: -90.08
## is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```