ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ

Μεταπτυχιακό Πρόγραμμα Συμπληρωματικής Ειδίκευσης
Εφαρμοσμένη Στατιστική

Μάθημα: Εφαρμοσμένη Μπεϋζιανή Στατιστική
Applied Bayesian Statistics

Καθηγητής: Ντζούφρας Ιωάννης

Τελική Εργασία

Κωνσταντίνα Τσάμη

ΠΕΡΙΕΧΟΜΕΝΑ

Περιγραφή της μελέτης / προβλήματος

Το πρόβλημά μας αφορά έρευνα που πραγματοποιήθηκε με στόχο να περιγραφούν οι παράγοντες που επιδρούν στην ετήσια δαπάνη καταναλωτών σε αγορές στα super markets. Αυτή η πληροφορία θα φανεί χρήσιμη στην διοίκηση των καταστημάτων κάποιας αλυσίδας super-market, αποφασίζοντας για κάποιες αλλαγές στον στρατηγικό σχεδιασμό που θα επιφέρουν μεγαλύτερη δαπάνη των καταναλωτών και έσοδα για την επιχείρηση.

Τα δεδομένα που έχουμε στη διάθεσή μας αφορούν τις παρακάτω μεταβλητές και περιέχουν 637 παρατηρήσεις.

Μεταβλητή απόκρισης ή μεταβλητή ενδιαφέροντός μας, η μέση ετήσια δαπάνη του καταναλωτή σε $: **spen_yr**.

Οι υπόλοιπες μεταβλητές που έχουμε στη διάθεση μας είναι όλες κατηγορικές και αφορούν τα παρακάτω.

- **Car**: αν ο καταναλωτής χρησιμοποιεί αμάξι ή όχι για να πάει στο super market (ΝΑΙ=1, ΟΧΙ=0)
- **Distance**: η απόσταση που εκτιμάει ο καταναλωτής ότι βρίσκεται το κατάστημα από το σπίτι του (1="<1 μίλι", 2= "1-5 μίλια", 3= "5-10 μίλια", 4= “>10 μίλια”).
- **Prices:** Εκτίμηση επιπέδου τιμών σύμφωνα με τον καταναλωτή (1=πολύ υψηλές, 2=υψηλές, 3=μέτριο επίπεδο τιμών, 4= χαμηλές, 5= πολύ χαμηλές).
- **Qfresh:** Εκτίμηση της ποιότητας των φρέσκων προϊόντων από τον καταναλωτή (1=πολύ χαμηλή, 2= χαμηλή, 3= μέτρια, 4= υψηλή, 5= πολύ υψηλή).
- **Qpack:** Εκτίμηση της ποιότητας των συσκευασμένων προϊόντων από τον καταναλωτή (1=πολύ χαμηλή, 2= χαμηλή, 3= μέτρια, 4= υψηλή, 5= πολύ υψηλή).
- **Queues:** Εκτίμηση της ουράς στα ταμεία (1=πάρα πολύ, 2= πολύ, 3= μέτρια ουρά, 4= μικρή, 5= πολύ μικρή).
- **Shop:** Αλυσίδα super market σε κατάστημα της οποίας συνηθίζει να ψωνίζει ο καταναλωτής (αποτελείται από κωδικοποίηση για 18 μεγάλες αλυσίδες).
- **sex:** το φύλο του καταναλωτή (0=γυναίκα, 1=άντρας).

Μας ενδιαφέρει λοιπόν να βρούμε ποιοι από τους παράγοντες που αναφέραμε επιδρούν σημαντικά και με ποιόν τρόπο στην μεταβλητή ενδιαφέροντός μας.

Ανάλυση Διακύμανσης (ANOVA)

Αρχικά, θέλουμε να εξετάσουμε την ύπαρξη διαφορών στο ποσό των χρημάτων που ξοδεύεται για αγορές στο supermarket μεταξύ των δύο φύλων αλλά και των διαφορετικών εκτιμώμενων επιπέδων τιμών. Για αυτό το σκοπό θα προσαρμόσουμε κάποια μοντέλα **one-way** και **tow-way anova** στο OpenBUGS.

Πιο συγκεκριμένα, θα προσαρμόσουμε ένα μοντέλο *tow-way anova* το οποίο θα περιλαμβάνει δύο διαφορετικά μοντέλα. Ένα μοντέλο για τις κύριες επιδράσεις (main effects)

$$\mu_{\alpha\beta} = \mu_o + \alpha_\alpha + \beta_2 + \beta_3 + \beta_4 + \beta_5$$

και ένα μοντέλο για την αλληλεπίδραση του φύλου και του επιπέδου τιμών (interaction term).

$$\mu_{\alpha\beta} = \mu_o + \alpha_\alpha + \beta_1 + \beta_2 + \beta_3 + \alpha\beta_{\alpha 2} + \alpha\beta_{\alpha 3} + \alpha\beta_{\alpha 4} + \alpha\beta_{\alpha 5}$$ [1]

Επιπρόσθετα, θα προσαρμόσουμε δύο ξεχωριστά μοντέλα *one-way anova* , το ένα θα αφορά την επίδραση του φύλου αποκλειστικά και το άλλο την επίδραση του εκτιμώμενου επιπέδου τιμών, καθώς επίσης και ένα σταθερό μοντέλο για να τα συγκρίνουμε (για τη διαδικασία *βλέπε ΠΑΡΑΡΤΗΜΑ, ΕΝΤΟΛΕΣ ΓΙΑ ΤΗΝ R ΚΑΙ ΤΟ WINBUGS, σελ. 10-19* , για τα διαγράμματα και τους διαγνωστικούς ελέγχους σύγκλισης *βλέπε ΠΑΡΑΡΤΗΜΑ, ΔΙΑΓΡΑΜΜΑΤΑ, Διαγράμματα 1-4* και *ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακες 1-3*).

Πίνακας 1*: Σύγκριση των διαφορετικών μοντέλων με το κριτήριο DIC*

Μοντέλο		**DIC**
$\mu_{\alpha\beta} = \mu_o$	$\boldsymbol{\mu_0}$	**10270**
$\mu_{\alpha\beta} = \mu_o + \alpha_\alpha$	$\boldsymbol{\mu_\alpha}$	10260
$\mu_{\alpha\beta} = \mu_o + \beta_2 + \beta_3 + \beta_4 + \beta_5$	$\boldsymbol{\mu_\beta}$	10230
$\mu_{\alpha\beta} = \mu_o + \alpha_\alpha + \beta_2 + \beta_3 + \beta_4 + \beta_5$	(main effects) $\boldsymbol{\mu_{\alpha\beta}}$	**10210**
$\mu_{\alpha\beta} = \mu_o + \alpha_\alpha + \beta_1 + \beta_2 + \beta_3 + \alpha\beta_{\alpha 2} + \alpha\beta_{\alpha 3} + \alpha\beta_{\alpha 4} + \alpha\beta_{\alpha 5}$	(int. term) $\boldsymbol{\mu_{\alpha\beta'}}$	10210

Οι posterior κατανομές των παραμέτρων είναι συμμετρικές όπως διαπιστώνουμε από τα density plots *(βλέπε ΠΑΡΑΡΤΗΜΑ, ΔΙΑΓΡΑΜΜΑΤΑ, Διαγράμματα 5.α-5.γ),* κάτι που μας επιτρέπει να χρησιμοποιήσουμε τον κλασσικό τύπο του κριτηρίου DIC και να εξάγουμε τα αποτελέσματα του *Πίνακα 1*.

Όπως παρατηρούμε, η αλληλεπιδράσεις μεταξύ φύλου και εκτιμώμενων επιπέδων τιμών είναι μη στατιστικά σημαντικές (DIC $\mu_{\alpha\beta}$ (**main effects**) = DIC $\mu_{\alpha\beta'}$ (int. term) = 10210). Από την άλλη, οι επιδράσεις του φύλου και του επιπέδου τιμών κρίνονται στατιστικά σημαντικές από τη στιγμή που η τιμή του DIC στα αντίστοιχα μοντέλα (DIC $\mu_{\alpha\beta}$ (main effects) = 10210, DIC μ_α = 10260, DIC μ_β = 10230) είναι μικρότερη από την τιμή του DIC του μηδενικού μοντέλου (DIC μ_0 = 10270) .

---

1 α_0 : η επίδραση του επιπέδου "άνδρας", β_2: η επίδραση του εκτιμώμενου επιπέδου τιμών 2, β_3: η επίδραση του εκτιμώμενου επιπέδου τιμών 3 κοκ. , $\alpha\beta_{\alpha 2}$:: η αλληλεπίδραση του επιπέδου "άνδρας" και επίπεδο τιμών 2, $\alpha\beta_{\alpha 3}$: η αλληλεπίδραση του επιπέδου "άνδρας" και επίπεδο τιμών 3 κοκ.

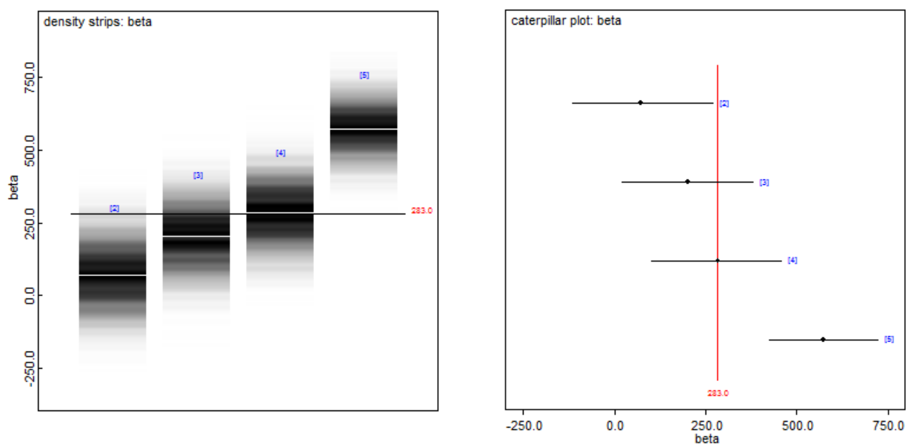
Για να ερευνήσουμε τον τρόπο με τον οποίο διαφοροποιείται το αναμενόμενο ποσό χρημάτων που δαπανούν οι καταναλωτές ανάμεσα στα δύο φύλα αλλά και μεταξύ των διαφορετικών εκτιμώμενων επιπέδων τιμών, θα εξετάσουμε τα box-plots *(βλέπε ΠΑΡΑΡΤΗΜΑ, ΔΙΑΓΡΑΜΜΑΤΑ, Διαγράμματα 5.α-5.γ)* και τα περιγραφικά μέτρα του μοντέλου των κύριων επιδράσεων (main effects) *(βλέπε ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακες 1-3)* :

$$\mu_{\alpha\beta} = 1168 + 245.1_{\alpha} + 71.45_{2} + 200.9_{3} + 281.9_{4} + 575.7_{5}$$

Από τον πίνακα των περιγραφικών μέτρων παρατηρούμε ότι ο μέσος και η διάμεσος των posterior κατανομών σχεδόν ταυτίζονται (π.χ alpha[2], mean=245.1 και median=245.5), ένδειξη συμμετρίας και καταλληλότητας του μέσου ως μέτρο κεντρικής θέσης. Παρατηρούμε επίσης μεγάλη τυπική απόκλιση για το μοντέλο μας (s, mean=1829) και ως εκ τούτου μεγάλο σφάλμα πρόβλεψης. Θα λέγαμε πως κάτι τέτοιο ήταν αναμενόμενο, αφού δεν θα ήταν λογικό το φύλο και το επίπεδο τιμών να είναι ικανά να προβλέψουν το αναμενόμενο χρηματικό ποσό που δαπανάται, είναι όμως ικανοί παράγοντες επίδρασης όπως διαπιστώνουμε. Επιπρόσθετα, το σφάλμα που οφείλεται στην προσομοίωση Monte Carlo είναι αρκετά μικρό, που σημαίνει ότι ο αλγόριθμός μας λειτουργεί με αρκετά καλή υπολογιστική ακρίβεια.

Τώρα, όσον αφορά τις παραμέτρους του μοντέλου παρατηρούμε ότι όλες είναι στατιστικά σημαντικές καθώς το 0 δεν εμπεριέχεται στα διαστήματα εμπιστοσύνης (**val2.5pc**, **val97.5pc** >0), με εξαίρεση την παράμετρο $\boldsymbol{\beta_2}$=**71.45** (beta[2], **val2.5pc=**-111, , **val97.5pc**=273.9 και **p.beta[2]**=0.759 ,όχι μεγάλη απόκλιση από το 0). Επίσης, από τα density strips και caterpillar plot *(Διάγραμμα 1)* παρατηρούμε ότι το 0 είναι σχετικά κεντρικό στην κατανομή της β_2. Από τα διαγράμματα αυτά επιπλέον, μπορούμε να πούμε πως όσο αυξάνεται το εκτιμώμενο επίπεδο των τιμών (πιο χαμηλές τιμές) τόσο αυξάνεται και η αναμενόμενη καταναλωτική δαπάνη. Κυρίως, διαφοροποιείται η posterior κατανομή του επιπέδου 5 "πολύ χαμηλές τιμές" σε σχέση με τα υπόλοιπα που παρουσιάζουν μεγάλη επικάλυψη (ειδικά τα επίπεδα τιμών 3 και 4, αναμένουμε να μην διαφοροποιούνται σημαντικά μεταξύ τους).

Διάγραμμα 1: *densuty strips και caterpillar plot των παραμέτρων $\beta_2, \beta_3, \beta_4, \beta_5$.*



Γενικότερα, οι άνδρες αναμένεται να ξοδεύουν περισσότερο από τις γυναίκες (θετικά διαστήματα εμπιστοσύνης, **val2.5pc=**69.11, , **val97.5pc**=419.9) κατά 245 \$ όταν το επίπεδο των τιμών εκτιμάται το ίδιο. Όταν το επίπεδο τιμών εκτιμάται "πολύ υψηλό" οι γυναίκες αναμένεται να δαπανούν 1.168 \$ για τις αγορές τους στο supermarket (ίδιο επιπ. τιμών). Όταν το επίπεδο τιμών εκτιμάται στο μέσο(3) "μέτριο επίπεδο τιμών", οι γυναίκες αναμένεται να ξοδεύουν 200 \$ παραπάνω, δηλαδή 1.168+200=1.368 \$ ετησίως. Αν τώρα, οι τιμές στο κατάστημα εκτιμούνται από μια γυναίκα ως "χαμηλές", τότε αναμένεται να ξοδέψει 282 \$ παραπάνω σε σχέση με το αν οι τιμές θεωρούνταν "πολύ υψηλές"(μικρή διαφορά από το επίπεδο 3). Τέλος, αν οι τιμές είναι "πολύ χαμηλές" σύμφωνα με την εκτίμηση μιας γυναίκας τα χρήματα που θα ξόδευε αναμένεται να είναι 576 \$ παραπάνω σε σχέση με το αν εκτιμούσε πως οι τιμές είναι "πολύ υψηλές".

Ερμηνευτικά ή προβλεπτικά μοντέλα

Στο σημείο αυτό, στόχος μας είναι να διερευνήσουμε μοντέλα ικανά να περιγράψουν όλους εκείνους τους παράγοντες που επιδρούν στην μεταβλητή ενδιαφέροντός μας (ετήσια δαπάνη καταναλωτών για αγορές σε supermarket). Προς αυτή τη κατεύθυνση θα κινηθούμε προσαρμόζοντας διάφορα μοντέλα και εξετάζοντάς τα για την ικανότητά τους να προβλέψουν σε ένα βαθμό, αλλά κυρίως να ερμηνεύσουν την ετήσια δαπάνη των καταναλωτών για αγορές από supermarket.

Αρχικά, εφαρμόζουμε ένα μοντέλο κατηγορικών συμμεταβλητών μόνο με τις κύριες επιδράσεις και ένα στο οποίο ενσωματώνουμε την αλληλεπίδραση της χρήσης αυτοκινήτου με την απόσταση, την αλληλεπίδραση του φύλου με τη χρήση αυτοκινήτου και την αλληλεπίδραση του φύλου με την απόσταση (*βλέπε ΠΑΡΑΡΤΗΜΑ, ΕΝΤΟΛΕΣ ΓΙΑ ΤΗΝ R ΚΑΙ ΤΟ WINBUGS, σελ 20-25). ,ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακας 5), έχοντας ενσωματώσει επιμέρους μοντέλα για τον υπολογισμό ελλιπων τιμών στις επεξηγηματικές μεταβήτές* . Δεν αναμένουμε αυτές οι αλληλεπιδράσεις να είναι σημαντικές., όπως επιβεβαιώνουμε συγκρίνοντας τις τιμές του κριτηρίου DIC στα δύο μοντέλα. (τύπος του **"Gelman"** $DIC=0.5[sd(devance)]^2 + mean(deviance)$, μοντέλο κύριων επιδράσεων **DIC=20539** , μοντέλο αλληλεπιδράσεων DIC=20538 , για τις deviances των μοντέλων *βλέπε ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακες 6-7*). Η διαφορά ανάμεσα στα δύο μοντέλα είναι ίση με 1 μονάδα, η διαφορά κρίνεται πολύ μικρή και τα δύο μοντέλα κρίνουμε ότι έχουν περίπου όμοια προβλεπτική ικανότητα.

Σειρά, τώρα, έχει να ενσωματώσουμε την αλληλεπίδραση της αλυσίδας supermarket (shop) με ποιοτικά χαρακτηριστικά όπως η ποιότητα των φρέσκων προϊόντων (Qfresh), η ποιότητα των συσκευασμένων προϊόντων (Qpack), το εκτιμώμενο επίπεδο τιμών (prices) και η ουρά στα ταμεία (Queues).(για λεπτομέρειες *βλέπε ΠΑΡΑΡΤΗΜΑ, ΕΝΤΟΛΕΣ ΓΙΑ ΤΗΝ R ΚΑΙ ΤΟ WINBUGS, σελ.25-30) ,ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακας 8)*. Παρατηρούμε ότι το DIC για αυτό το μοντέλο είναι κατά 7 μονάδες μεγαλύτερο απο εκείνο των κύριων επιδράσεων (DIC=20546, *βλέπε ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακας 9)*, συνεπώς η προσαρμογή του είναι χειρότερη από εκείνη των κυρίων επιδράσεων. Αφαιρούμε τους όρους αλληλεπίδρασης ένα - ένα και ελέγχουμε ότι το DIC είναι μεγαλύτερο από εκείνο των κυρίων επιδράσεων.

Σαν επόμενο βήμα, θα δοκιμάσουμε τη προσαρμογή ενός μοντέλου "random effects", υποθέτοντας ενός είδους συσχέτιση στους συντελεστές επίδρασης της αλυσίδας supermarket. Αυτό υποθέτει ότι οι διαφορές που παρατηρούνται μεταξύ των διαφορετικών καταστημάτων είναι κάτω από ένα συγκεκριμένο πρίσμα με συγκεκριμένη μεταβλητότητα (*βλέπε ΠΑΡΑΡΤΗΜΑ, ΕΝΤΟΛΕΣ ΓΙΑ ΤΗΝ R ΚΑΙ ΤΟ WINBUGS, σελ.30-31)* . Βλέπουμε ωστόσο ότι το μοντέλο δεν έχει τόσο καλή προσαρμογή (DIC=20552 > 20539, *βλέπε ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακας 10)*.

Στη συνέχεια θα ελέγξουμε την σημαντικότητα της μεταβλητής "shop" προσαρμόζοντας ένα μοντέλο κυρίων επιδράσεων (όπως προηγουμένως), χωρίς τη συγκεκριμένη μεταβλητή (*βλέπε ΠΑΡΑΡΤΗΜΑ, ΕΝΤΟΛΕΣ ΓΙΑ ΤΗΝ R ΚΑΙ ΤΟ WINBUGS, σελ.31-32)*. Η τιμή του κριτηρίου DIC του μοντέλου είναι ίση με **18230** (*βλέπε ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακας 11)*, κατά πολύ μικρότερη όλων των μοντέλων που έχουμε προσαρμόσει μέχρι τώρα. Κρίνουμε ,λοιπόν, πως η επίδραση της μεταβλητής "shop" δεν είναι στατιστικά σημαντική.

Ακόμα μία προσέγγιση που θα μπορούσαμε να εξετάσουμε είναι η διαχείρηση των μεταβλητών ταξινόμησης ("ordinal") όπως το επίπεδο τιμών (prices), η ποιότητα των φρέσκων προϊόντων κλπ, ως ποσοτικών μεταβλητών (*βλέπε ΠΑΡΑΡΤΗΜΑ, ΕΝΤΟΛΕΣ ΓΙΑ ΤΗΝ R ΚΑΙ ΤΟ WINBUGS, σελ. 33)*. Μία τέτοια προσέγγιση δίνει υψηλότερη τιμή DIC από το προηγούμενο μοντέλο (DIC=18424 > **18230**, (*βλέπε ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακας 12).* Επομένως, θα μείνουμε στο μοντέλο που προσαρμόσαμε στο προηγούμενο βήμα.

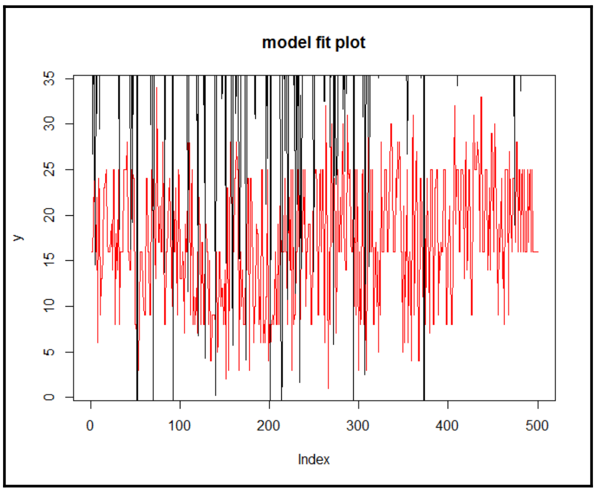
Σε γενικές γραμμές οι μεταβλητές που φαίνεται να είναι στατιστικά σημαντικές είναι οι car, sex, distance, για το 2ο και 4ο επίπεδο, Qfresh, Qpack, Queues, οι teleyta;iew για τα επίπεδα 1 και 5. Αυτό το συμπεραίνουμε από τον πίνακα περιγραφικών μέτρων των posterior κατανομών των παραμέτρων (*βλέπε ΠΑΡΑΡΤΗΜΑ, ΠΙΝΑΚΕΣ, Πίνακες 13-14).* Βλέπουμε ότι το 0 δεν περιλαμβάνεται στα διαστήματα εμπιστοσύνης των αντίστοιχων παραμέτρων των μεταβλητών (alpha1[2], alpha2[2] , alpha3[2], alpha3[4], alpha4[1], alpha4[5], alpha5[1], alpha5[5], alpha6[1], alpha6[5], alpha7[1], alpha7[5]). Η μεταβλητή Queues φαίνεται στατιστικά σημαντική αλλά με μικρότερη πιθανότητα. Σε αρκετά υψηλό βαθμό σημαντικές φαίνονται οι εξής μεταβλητές: το πρώτο επίπεδο τιμών (πολύ υψηλές) (alpha6[1]), το 5ο επίπεδο της ποιότητας των συσκευασμένων προϊόντων (alpha5[5]), το 5ο επίπεδο και το 1ο επίπεδο των φρέσκων προϊόντων(alpha4[1], alpha4[1]), το 2ο επίπεδο της απόστασης (alpha3[2]) και η χρήση αυτοκινήτου(alpha1[1],alpha2[2]), για τις αντίστοιχες παραμέτρους παρατηρούμε ότι το 0 έχει μεγάλη απόσταση από τα διαστήματα εμπιστοσύνης, και το p0 ισούται με την μονάδα.

Δίνοντας μία ερμηνεία για τις παραμέτρους από τους posterior μέσους των παραμέτρωνν θα λέγαμε επιγραμματικά τα εξής:

- Η χρήση αυτοκινήτου αυξάνει την a-posteriori αναμενόμενη δαπάνη των καταναλωτών για αγορές στα supermarket, κατά 304$ (304$ λιγότερο για μη χρήση) σε σχέση με τον ολικό μέσο, σε άτομα του ίδιου φύλου, που εκτιμούν τις τιμές στο ίδιο επίπεδο, την απόσταση το ίδιο, την ποιότητα των φρέσκων και συσκευασμένων προϊόντων και την ουρά στα ταμεία επίσης στο ίδιο επίπεδο.
- Όταν η ποιότητα των φρέσκων προϊόντων εκτιμάται ως πολύ καλή, τότε αναμένουμε a-posteriori να ξοδεύει κάποιος 398.9$ παραπάνω σε σχέση με τον ολικό μέσο, όταν τα υπόλοιπα χαρακτηρίστικά εκτιμούνται στο ίδιο επίπεδο.

Όταν έχουμε ένα τυπικό προφίλ καταναλωτή η ετήσια δαπάνη για αγορές σε supermarket αναμένεται να είναι a-posteriori ίση με **802 $** (m, mean) **,** με 95% πιθανότητα να ξοδεύει από 639$ έως 959$.
Πρέπει να σημειώσουμε σε αυτό το σημείο ότι το σφάλμα πρόβλεψης του μοντέλου μας είναι αρκετά μεγάλο (s= 1713 $) και η προσαρμογή του μοντέλου όπως φαίνεται στο *Διάγραμμα 2* δεν είναι ικανοποιητική.

Διάγραμμα 2: *Διάγραμμα προσαρμογής του μοντέλου. Με κόκκινο χρώμα εμφανίζονται οι πραγματικές τιμές (spen_yr) και με μαύρο χρώμα οι εκτιμώμενες (mu)*



Επιλογή μεταβλητών

Αφού καταλήξαμε σε ένα μοντέλο ανάμεσα σε διαφορετικά μοντέλα διαφορετικών προσεγγίσεων, θα προχωρήσουμε με το κομμάτι της επιλογής μεταβλητών με το κριτήριο της ερμηνευτικότητας που κυρίως μας ενδιαφέρει στην παρούσα εργασία. Θέλουμε λοιπόν να επιλέξουμε τον μικρότερο δυνατό αριθμό μεταβλητών (μεγαλύτερη ποινή ως προς τον αριθμό μεταβλητών) με την καλύτερη δυνατή ερμηνευτική και προβλεπτική ικανότητα. Ένα τέτοιο μοντέλο έχει στόχο να μας βοηθήσει στο ποιες μεταβλητές και ποιος συνδυασμός μεταβλητών μας εξηγούνμε τον καλύτερο δυνατό τρόπο την μεταβλητότητα της εξαρτημένης μας μεταβλητής (ετήσια δαπάνη καταναλωτών για αγορές στα supermarkets).

Ξεκινάμε χρησιμοποιώντας το πακέτο “BAS” της R (για τις εντολές *βλέπε ΠΑΡΑΡΤΗΜΑ, ΕΝΤΟΛΕΣ ΓΙΑ ΤΗΝ R ΚΑΙ ΤΟ WINBUGS, σελ.34-36*). Από τα αποτελέσματα για τις διαφορετικές “prior” που χρησιμοποιήσαμε εξήγαμε τα καλύτερα μοντέλα σύμφωνα με διαφορετικές μεθόδους (MAP, MP). Τα μοντέλα αυτά περιγράφονται στον *Πίνακα 2* που ακολουθεί.

Πίνακας 2*: Σύγκριση των διαφορετικών μοντέλων μέσω του Variable Selection στην R.*

	Μεταβλητές	
	MP [2]	**MAP** [3]
Hyper-g prior Βήτα Διωνυμική	**Dcar**	Dsex
Hyper-g prior Ομοιόμορφη	**Dcar** + Dprices4 + DQfresh3 + **DQfresh5**	**Dcar** + Dprices4 + DQfresh3 + **DQfresh5** + DQueues3
g- prior Βήτα-Διωνυμική	**Dcar**	Dsex
g- prior Ομοιόμορφη	**Dcar**	**Dcar**
BIC Βήτα-Διωνυμική	**Dcar**	Dsex
BIC Ομοιόμορφη	**Dcar**	**Dsex**

Όταν χρησιμοποιούμε Βήτα-Διωνυμική οδηγούμαστε σε πιο απλά μοντέλα σε σχέση με την Ομοιόμορφη η οποία αυξάνει τα p-values, τα postrior odds και τα inclusion probabilities των μεταβλητών. Σε πιο απλό μοντέλο επίσης, μας οδηγεί το κριτήριο του Median Probability. Παρατηρώ επιπλέον ότι η g-prior συμπεριφέρεται όπως το BIC, μας οδήγησε στα ίδια μοντέλα. Σε γενικές γραμμές το μοντέλο που περιέχει μόνο την κατηγορική μεταβλητή της χρήσης αυτοκινήτου ή όχι (Dcar) φαίνεται να υπερισχύει σε όλες τις διαφορετικές προσεγγίσεις, είμαστε λοιπόν πιο σίγουροι ότι αυτή η μεταβλητή είναι σημαντική και πρέπει να εισαχθεί στο μοντέλο μας.

2 Median Probability Model, περιέχει μεταβλητές με inclusion probabilities > 0.5
3 Maximum a-posteriori Model, μοντέλο με τη μεγαλύτερη posterior πιθανότητα

Σε αυτό το σημείο, επαναλαμβάνουμε την διαδικασία μέσω του OpenBUGS (για τις εντολές *βλέπε ΠΑΡΑΡΤΗΜΑ, ΕΝΤΟΛΕΣ ΓΙΑ ΤΗΝ R ΚΑΙ ΤΟ WINBUGS, σελ.37-41*). Όπως παρατηρούμε στον *Πίνακα 3*, οδηγούμαστε σε πιο πολύπλοκα μοντέλα μέσω του OpenBUGS. Όσον αφορά την παράμετρο σύγκλισης (G) παρατηρούμε ότι η τιμή της είναι μεγαλύτερη στην **hyper-g** prior με **Βήτα-Διωνυμική** κατανομή, που σημαίνει ότι το μοντέλο μας σε αυτή τη περίπτωση είναι καλύτερο (το ίδιο ισχύει και υπολογίζοντάς το στην R).

Επίσης, παρατηρούμε ότι και εδώ η μεταβλητή Dcar βρίσκεται σε όλα τα μοντέλα που πήραμε από όλες τις διαφορετικές priors. Είμαστε βέβαιοι λοιπόν, ότι αυτή η μεταβλητή είναι σημαντική. Την μεταβλητή DQfresh5 τη συναντάμε στα περισσότερα μοντέλα του OpenBUGS καθώς και στην hyper g-prior με Ομοιόμορφη κατανομή στην R και τη μεταβλητή Qprices5 στα περισσότερα μοντέλα του OpenBUGS.

Για τις τελευταίες δύο μεταβλητές μπορούμε να πούμε πως υπάρχει υψηλή πιθανότητα να επιδρούν στην μεταβλητή μας δεν είμαστε σε θέση όμως να πούμε με βεβειότητα. Φαίνεται πάντως, από τα μέχρι τώρα δεδομένα, πως η ποιότητα των φρέσκων προϊόντων και η τιμή φαίνονται να παίζουν σημαντικό ρόλο στις όταν βρίσκονται στις μέγιστες τιμές τους. Δηλαδή, όταν οι τιμές εκτιμάται ότι είναι πολύ χαμηλές ή η ποιότητα των φρέσκων προϊόντων πολύ καλή, τότε τόσο περισσότερα χρήματα αναμένεται να ξοδευτούν (θετική επίδραση στη μεταβλητή ενδιαφέροντος, με βάση και τα προηγούμενα ευρήματα). Θα λέγαμε ότι κάτι τέτοιο είναι λογικό, διότι κάποιος που εκτιμά ότι οι τιμές ενός καταστήματος είναι πολύ οικονομικές ή είναι πολύ ικανοποιημένος από την ποιότητα των φρέσκων προϊόντων αναμένεται να ξοδέψει με περισσότερη ευκολία.

Πίνακας 3*: Σύγκριση των διαφορετικών μοντέλων μέσω του V S στο OpenBUGS.*

	Μεταβλητές - MP [4]	**G**[5]
Hyper-g prior Βήτα Διωνυμική	**Dcar** + **DQfresh5** + Dprices2 + **Dprices5**	4.868
Hyper-g prior Ομοιόμορφη	**Dcar** + DQfresh3 + **DQfresh5**+ **Dprices5** + DQpack3	3.736
g- prior Βήτα-Διωνυμική	**Dcar** + DQfresh3 + **DQfresh5**+ Dprices2 + **Dprices5** + DQpack5	n
g- prior Ομοιόμορφη	**Dcar**	n
Unit Information empirical Βήτα-Διωνυμική	Όλες εκτός της Dsex	-
Unit Information empirical Ομοιόμορφη	**Dcar +** **DQfresh5** + Dprices2 + **Dprices5** + DQpack5 + Dqueues3 + DQueues4	-

4 Μεταβλητές με inclusion probability >0.5 (mean(g)>0.5 και median(g)=1 στον πίνακα αποτελεσμάτων του μοντέλου)

5 BMA εκτίμηση του shrinkage parameter μέσω OpenBUGS (median τιμή)

Συμπεράσματα

Ανακεφαλαιώνοντας, από την ανάλυσή μας έως τώρα θα λέγαμε ότι ο παράγοντας που είμαστε βέβαιοι ότι επιδρά σημαντικά στο ποσό που ξοδεύουν οι καταναλωτές για τις αγορές τους από το supermarket είναι η χρήση αυτοκινήτου. Αυτό σημαίνει ότι αν οι καταναλωτές χρησιμοποιούν το αυτοκίνητό τους για να πάνε στο supermarket αναμένεται να ξοδεύουν σημαντικά περισσότερα χρήματα σε σχέση με όσους δεν τον χρησιμοποιούν. Αυτό που ενδεχομένως θα μπορούσε να σκεφτεί η διοίκηση ενός καταστήματος θα ήταν να κάνει την πρόσβαση πιο εύκολη στους οδηγούς (περισσότερες θέσεις parking κλπ).

Επισφαλείς είμαστε ως προς την σημαντικότητα του παράγοντα της τιμής και της ποιότητας των φρέσκων προϊόντων. Σε γενικές γραμμές φαίνεται όμως να επιδρούν θετικά. Όσο χαμηλότερες εκτιμούνται οι τιμές από τους καταναλωτές τόσο περισσότερα χρήματα αναμένεται να ξοδεύουν, το ίδιο ισχύει και με την ποιότητα των φρέσκων προϊόντων, όσο καλύτερη ποιότητα φρέσκων προϊόντων διαθέτει ένα κατάστημα τόσο περισσότερα χρήματα αναμένεται να δαπανηθούν.

Σαν περαιτέρω ανάλυση και εκτίμηση του καλύτερου περιγραφικού μοντέλου θα μπορούσαμε να τρέξουμε τα διαφορετικά μοντέλα με τις μεταβλητές που εξάγαμε μέσω των διαφορετικών προσεγγίσεων στο OpenBUGS και να τα συγκρίνουμε μέσω του κριτηρίου “DIC”. Για καλύτερη ερμηνευτική ικανότητα θα προτιμούσαμε το μοντέλο με το μικρότερο δυνατό DIC σε συνδυασμό με τον μικρότερο δυνατό αριθμό παραμέτρων.

Σε γενικές γραμμές, όλα αυτά περιέχουν ένα στοιχείο υποκειμενικότητας καθώς ερωτήθηκαν οι συμμετέχοντες ως προς το πως εκτιμούν τα στοιχεία αυτά. Σε πιο ασφαλή συμπεράσματα θα καταλήγαμε σχεδιάζοντας μια πιο εκτεταμένη πλην όμως κοστοβόρα έρευνα, ελέγχοντας και συγκρίνοντας το επίπεδο των τιμών και των προϊόντων μεταξύ των διαφορετικών καταστημάτων και επιλέγοντας καταναλωτές ενδεχομένως με βάση το εισόδημά τους και την οικογενειακή τους κατάσταση.

Εντολές για την R και το OpenBUGS

Αρχική επεξεργασία των δεδομένων στην R

Πριν ξεκινήσουμε με την προσαρμογή ενός μοντέλου, πρέπει να κάνουμε την απαραίτητη προεργασία των δεδομένων μας.

Αρχικά, διαβάζουμε τα δεδομένα μας στην **R**.

```
library(foreign)
sm<-read.spss("C://R//14_supermarkets.sav",use.value.label=TRUE,to.d
ata.frame=TRUE)
```

Σύντομη περιγραφή των μεταβλητών.

```
View(sm)
head(sm)
str(sm)
```

Ομαδοποιούμε τις κατηγορίες της μεταβλητής shop με λίγες παρατηρήσεις.

```
x<-summary(sm$shop)
x<-x[x<=10]
x<-names(x)
sm$shop<-as.character(sm$shop)

for (j in 1:length(x)) { for (i in 1:dim(sm)[1]){
  if (is.na(sm$shop[i])==FALSE){
   if (sm$shop[i]==x[j]){ sm$shop[i]<-"Others"}
  }
}}

sm$shop<-as.factor(sm$shop)
rm(x)
```

Στη συνέχεια μετατρέπουμε όλα τα δεδομένα σε ποσοτικές μεταβλητές με την κατάλληλη κωδικοποίηση, ώστε να μπορούμε να τα εισάγουμε στο “OpenBUGS”.

```
attach(sm)
spen_yr<-as.character(spen_yr)
spen_yr<-as.numeric(spen_yr)

car<-as.numeric(car)
car[car==1]<-0
car[car==2]<-1

shop<-as.numeric(shop)
```

```r
distance<-as.numeric(distance)

sex<-as.numeric(sex)
sex[sex==1]<-0
sex[sex==2]<-1
```

Υπολογίζουμε το πλήθος των γραμμών με ελλιπείς τιμές στις επεξηγηματικές μεταβλητές. Οι ελλιπείς τιμές στην μεταβλητή απόκρισης δεν μας απασχολούν καθώς θα εκτιμηθούν αυτόματα από το μοντέλο μας.

```r
x<-apply(sm[,-match("spen_yr",colnames(sm))],1,function(x)
sum(is.na(x)))
sum(x>0) # NA
## [1] 91
```

Έχουμε 91 γραμμές με missing data από τις 637 στο σύνολο.
Μπορούμε εάν θέλουμε να υπολογίσουμε αυτά τα missing values μοντελοποιώντας τις συμμεταβλητές στο “OpenBUGS”.

Ανάλυση Διακύμανσης (ANOVA)

Μετατρέπουμε τα δεδομένα που θα χρειαστούμε σε μορφή λίστας και στη συνέχεια τα εισάγουμε στο OpenBUGS.

```r
sm.na <-na.omit(cbind(sm$spen_yr,sm$sex,sm$prices))
dput(list(spen_yr=sm.na[,1],X=t(cbind(sm.na[,2],sm.na[,3]))))
rm(sm.na)
```

Τα δεδομένα μας είναι διαθέσιμα στο παρακάτω αρχείο.

Για την εξέταση των διαφορών στο ποσό των χρημάτων που ξοδεύεται για αγορές στο supermarket μεταξύ των δύο φύλων αλλά και των διαφορετικών εκτιμώμενων επιπέδων τιμών, θα προσαρμόσουμε ένα μοντέλο **tow-way anova** στο OpenBUGS.

Πιο συγκεκριμένα, θα προσαρμόσουμε ένα μοντέλο το οποίο θα περιλαμβάνει πέντε διαφορετικά μοντέλα. Το μοντέλο που επιλέγουμε κάθε φορά το επιλέγουμε μετατρέποντας τα άλλα δύο σε σχόλιο. Θα τρέξουμε το μοντέλο για 2000 επαναλήψεις απορρίπτοντας τις 1000 πρώτες για τη σταθεροποίηση του αλγορίθμου.

```
model{
    # model's likelihood
    for (i in 1:n){
       mu[i] <- m + alpha[ X[i,1] +1 ] + beta[ X[i,2] ]
                                + ab[ X[i,2], X[i,1]+1 ]
#interaction term
       #mu[i] <- m + alpha[ X[i,1] +1 ] + beta[ X[i,2] ]   #main
effects
       #mu[i] <- m + alpha[ X[i,1] +1 ]                    #anova 1
       #mu[i] <- m + beta[ X[i,2] ]                        #anova 2
       #mu[i] <- m                                         #m0
       spen_yr[i] ~ dnorm( mu[i], tau )
    }


    #### CR Constraints
    alpha[1] <- 0.0
    beta[1] <- 0.0
    ab[1,1] <- 0.0
    ab[1,2] <- 0.0
    for (k in 2:LB){
    ab[k,1] <- 0.0
    }


    # priors
    m~dnorm( 0.0, 1.0E-04)
    alpha[2]~dnorm(0.0, 1.0E-04)
    for (k in 2:LB){
      beta[k]~dnorm(0.0, 1.0E-04)
    }
    for (a in 2:LA){
        for (b in 2:LB){
             ab[b,a]~dnorm( 0.0, 1.0E-04)
    }}
    tau ~dgamma( 0.01, 0.01)
    s <- sqrt(1/tau) # precission
    for (a in 1:LA){
      p.alpha[a] <- step(alpha[a])
    }
    for (b in 2:LB){
      p.beta[b] <- step(beta[b])
    }
    for (a in 2:LA){ for (b in 2:LB) {
      p.ab[b,a] <- step(ab[b,a])
    }}


    #pseudo-R
    sy2<-pow(sd(spen_yr[]),2)
    s2<-1/tau
    R2B<-1-s2/sy2

}
```

```
INITS
list( m=1.0, alpha=c(NA,0), tau=1.0, beta=c(NA,0,0,0,0),
  ab=structure(.Data=c(NA,NA, NA, 0, NA, 0, NA, 0, NA,
0), .Dim=c(5,2)) )
```

Συνεχίζουμε τη διαδικασία μας στην R , μέσω του πακέτου “R2WinBUGS”.

Φορτώνουμε τις απαραίτητες βιβλιοθήκες

```
library(R2WinBUGS)
library(BRugs)
```

```
library(foreign)
sm<-read.spss("C://R//
14_supermarkets.sav",use.value.label=TRUE,to.data.frame=TRUE)


#Import data - sample size
sm.na<-na.omit(cbind(sm$spen_yr,sm$sex,sm$prices))
data <-list( spen_yr=sm.na[,1], X=cbind(sm.na[,2],sm.na[,3]), LA=2,
LB=5, n=572 )
rm(sm.na)

# initial values
# should be a list with one list for each chain
#
inits1<-list(
 list( m=1.0, alpha=c(NA,0), tau=1.0, beta=c(NA,0,0,0,0),
  ab=matrix(c(NA,NA, NA, 0, NA, 0, NA, 0, NA, 0),5, 2,byrow=TRUE )))

# defining the names of the parameters we wish to monitor
parameter.names <- c( 's', 'tau' , 'alpha', 'beta' , 'ab' , 'm' ,
'p.alpha' , 'p.beta', 'p.ab' )


# defining the directory of OpenBUGS
openbugs.dir <- "C:/Program Files (x86)/OpenBUGS"

# generating random samples using OpenBUGS
model.ab.intr <-bugs( data, inits1, model.file =
                                    "anova_all_models.txt",
                    parameters = parameter.names,n.chains = 1,
                    n.iter = 2000, n.burnin=1000, n.thin=1,
                    bugs.directory = openbugs.dir, debug=F,
                    program="OpenBUGS")

# function for checking the centrality of zero
# 2013 April by Ioannis Ntzoufras
#
p0 <- function( bugs.object, digits=3){
```

```
  mcmc.output<-bugs.object$sims.matrix
  n.iter <- nrow(mcmc.output)
  n.par  <- ncol(mcmc.output)
  mcmc.output<-mcmc.output[ , -n.par]     #vgazei to Deviance
  temp<-apply( mcmc.output < 0, 2, mean)
  res <- pmin( temp, 1-temp)  #min gia kathe stoixeio (temp,1-temp)
  return( round(res,digits) )
}


# calculating the probability of zero to be central in the posterior
densities
p0(model.ab.intr)


# --------------------------------
# Plots without functions
# --------------------------------

#trace plot
head(model.ab.intr$sims.matrix)

# Trace plots of  "m", "tau", "alpha"
x<-
match(c("m","tau","alpha[2]"),colnames(model.ab.intr$sims.matrix))
par(mfrow=c(3,1))
for (i in x){
plot(model.ab.intr$sims.matrix[,i],type='l',
  main=paste("Trace plot of ",colnames(model.ab.intr$sims.matrix)
[i]),
    ,xlab="iterations",ylab=paste(colnames(model.ab.intr$sims.matrix)
[i]))
}

# Trace plots of  "beta"
par(mfrow=c(4,1))
for (i in 4:7){
plot(model.ab.intr$sims.matrix[,i],type='l',
  main=paste("Trace plot of ",colnames(model.ab.intr$sims.matrix)
[i]),
    ,xlab="iterations",ylab=paste(colnames(model.ab.intr$sims.matrix)
[i]))
}
```

```
# Trace plots of  "ab"
par(mfrow=c(4,1))
for (i in 8:11){
plot(model.ab.intr$sims.matrix[,i],type='l',
  main=paste("Trace plot of ",colnames(model.ab.intr$sims.matrix)
[i]),
    ,xlab="iterations",ylab=paste(colnames(model.ab.intr$sims.matrix)
[i]))
}

#acf plots

# Autocorrelation of "m", "tau", "alpha"
x<-
match(c("m","tau","alpha[2]"),colnames(model.ab.intr$sims.matrix))
par(mfrow=c(3,1))
for (i in x){
acf(model.ab.intr$sims.matrix[,i],main=paste("ACF plot of ",
              colnames(model.ab.intr$sims.matrix)[i] ),lag=100)
}

# Autocorrelation of "beta"
par(mfrow=c(4,1))
for (i in 4:7){
acf(model.ab.intr$sims.matrix[,i],main=paste("ACF plot of ",
              colnames(model.ab.intr$sims.matrix)[i] ),lag=100)
}

# Autocorrelation of "ab"
par(mfrow=c(4,1))
for (i in 8:11){
acf(model.ab.intr$sims.matrix[,i],main=paste("ACF plot of ",
              colnames(model.ab.intr$sims.matrix)[i] ),lag=100)
}


# --------------------------------
# Summarizing
# --------------------------------

#stats

X<-matrix(0,8,12)

for ( j in 1:12){
X[,j]<-rbind(c(summary(model.ab.intr$sims.matrix[,j]),
             quantile(model.ab.intr$sims.matrix[,j],probs=0.25),
    quantile(model.ab.intr$sims.matrix[,j],probs=0.95)))
}
X<-round(X,2)
x<-summary(model.ab.intr$sims.matrix[,1])
q1<-quantile(model.ab.intr$sims.matrix[,1],probs=0.25)
q2<-quantile(model.ab.intr$sims.matrix[,1],probs=0.95)
```

```
rownames(X)<-c(names(x),names(q1),names(q2));colnames(X)<-
colnames(model.ab.intr$sims.matrix)[1:12]
X

rm(q1,q2,x)

#density plots

# density plots of  "m", "tau", "alpha"
x<-
match(c("m","tau","alpha[2]"),colnames(model.ab.intr$sims.matrix))
par(mfrow=c(2,2))
for (i in x){
plot(density(model.ab.intr$sims.matrix[,i]),
         main=paste(colnames(model.ab.intr$sims.matrix)[i]))
}

# density plots of  "beta"
par(mfrow=c(2,2))
for (i in 4:7){
plot(density(model.ab.intr$sims.matrix[,i]),
         main=paste(colnames(model.ab.intr$sims.matrix)[i]))
}

# density plots of  "ab"
par(mfrow=c(2,2))
for (i in 8:11){
plot(density(model.ab.intr$sims.matrix[,i]),
         main=paste(colnames(model.ab.intr$sims.matrix)[i]))
}



# --------------------------------
# Ergotic mean plots
# --------------------------------

par(mfrow=c(3,1))

# Ergodic Trace plot of  "theta"
x<-model1.exp$sims.matrix[,"theta"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",
   ylab=colnames(model1.exp$sims.matrix)[1] )

# Ergodic Trace plots of  "m", "tau", "alpha"
x<-
match(c("m","tau","alpha[2]"),colnames(model.ab.intr$sims.matrix))
par(mfrow=c(3,1))
for (i in x){
x1<-model.ab.intr$sims.matrix[,i]
plot( cumsum(x1)/1:length(x1), type='l',xlab="iterations",
   ylab=paste(colnames(model.ab.intr$sims.matrix)[i] ))
}
# Ergodic Trace plots of  "beta"
```

```
par(mfrow=c(4,1))
```

```
for (i in 4:7){
x1<-model.ab.intr$sims.matrix[,i]
plot( cumsum(x1)/1:length(x1), type='l',xlab="iterations",
   ylab=paste(colnames(model.ab.intr$sims.matrix)[i] ))
}

# Ergodic Trace plots of  "ab"
par(mfrow=c(4,1))
for (i in 8:11){
x1<-model.ab.intr$sims.matrix[,i]
plot( cumsum(x1)/1:length(x1), type='l',xlab="iterations",
   ylab=paste(colnames(model.ab.intr$sims.matrix)[i] ))
}

# ---------------------------------
# compare models using Gelman's DIC
# ---------------------------------

##########anova a############
inits1<-list(
          list( m=1.0, alpha=c(NA,0), tau=1.0 ))
parameter.names <- c( 's', 'tau' , 'alpha', 'm' , 'p.alpha' )

model.a<- bugs( data, inits1, model.file = "anova_a.txt",
                    parameters = parameter.names,n.chains = 1,
             n.iter = 2000, n.burnin=1000, n.thin=1,
             bugs.directory = openbugs.dir, debug=F,
  program="OpenBUGS")



##########anova b############
inits1<-list(
          list( m=1.0, tau=1.0, beta=c(NA,0,0,0,0) ))
parameter.names <- c( 's', 'tau' , 'beta' , 'm' , 'p.beta' )

model.b<- bugs( data, inits1, model.file = "anova_b.txt",
       parameters = parameter.names,n.chains = 1,
         n.iter = 2000, n.burnin=1000, n.thin=1,
       bugs.directory = openbugs.dir, debug=F,
program="OpenBUGS")




##########anova ab###########
inits1<-list(
          list( m=1.0, alpha=c(NA,0), tau=1.0,
beta=c(NA,0,0,0,0) ))
parameter.names <- c( 's', 'tau' , 'alpha', 'beta' , 'm' , 'p.alpha'
          , 'p.beta'  )

model.a.b<- bugs( data, inits1, model.file = "anova_a_b.txt",
              parameters = parameter.names,n.chains = 1,
       n.iter = 2000, n.burnin=1000, n.thin=1,
```

```
       bugs.directory = openbugs.dir, debug=F,
program="OpenBUGS")


########## null ###########
inits1<-list(
          list( m=1.0, tau=1.0 ) )
parameter.names <- c( 's', 'tau' , 'm' )

# generating random samples using OpenBUGS
model.null<- bugs( data, inits1, model.file = "anova_null.txt",
              parameters = parameter.names,n.chains = 1,
       n.iter = 2000, n.burnin=1000, n.thin=1,
       bugs.directory = openbugs.dir, debug=F,
program="OpenBUGS")



###DIC - Gelman's formula

#model.a
0.5*var(model.a$sims.matrix[,"deviance"])
+mean(model.a$sims.matrix[,"deviance"])

#model.b
0.5*var(model.b$sims.matrix[,"deviance"])
+mean(model.b$sims.matrix[,"deviance"])

#model.a.b
0.5*var(model.a.b$sims.matrix[,"deviance"])
+mean(model.a.b$sims.matrix[,"deviance"])

#interaction model
0.5*var(model.ab.intr$sims.matrix[,"deviance"])
+mean(model.ab.intr$sims.matrix[,"deviance"])

#null model
0.5*var(model.null$sims.matrix[,"deviance"])
+mean(model.null$sims.matrix[,"deviance"])
```

Σαν τελευταίο βήμα θα εφαρμόσουμε βηματική μέθοδο “stepwise” για την επιλογή μεταβλητών-παραμέτρων, χρησιμοποιώντας το κριτήριο “DIC”.

Για να το κάνουμε αυτό, θα χρειαστεί να προσαρμόσουμε ένα νέο μοντέλο στο OpenBUGS όπου θα δημιουργήσουμε τις διαφορετικές προσομοιωμένες τιμές για κάθε μοντέλο που προκύπτει αφαιρώντας και προσθέτοντας μία-μία τις μεταβλητές.Αρχικά, θα ορίσουμε τα διαφορετικά μοντέλα μας σε διαφορετικές μεταβλητές και στη συνέχεια θα εξάγουμε την τιμή του κριτηρίου DIC για το κάθε ένα μέσω της αντίστοιχης επιλογής στο μενού του OpenBUGS.

Το μοντέλο μας λοιπόν θα έχει την παρακάτω σύνταξη.
Η conditional likelihood σε κάθε διαφορετικό μοντέλο θα ορίζεται ως εξής.

```
model{

    pi<-3.14
     # model's likelihood
     for (i in 1:n){
          y1[i] <- spen_yr[i]
          y2[i] <- spen_yr[i]
          y3[i] <- spen_yr[i]
          y4[i] <- spen_yr[i]
          y5[i] <- spen_yr[i]
          y1[i] ~ dnorm( mu[i,1], tau[1] )
          y2[i] ~ dnorm( mu[i,2], tau[2] )
          y3[i] ~ dnorm( mu[i,3], tau[3] )
          y4[i] ~ dnorm( mu[i,4], tau[4] )
          y5[i] ~ dnorm( mu[i,5], tau[5] )


       mu[i,1] <- m[1] + alpha[ X[i,1] +1 , 1 ] + beta[ X[i,2], 1 ]
                                + ab[ X[i,2], X[i,1]+1 ]
#interaction effect
       mu[i,2] <- m[2] + alpha[ X[i,1] +1, 2 ] + beta[ X[i,2], 2 ]
#main effect
       mu[i,3] <- m[3] + alpha[ X[i,1] +1, 3 ]
       mu[i,4] <- m[4] + beta[ X[i,2], 3 ]
       mu[i,5] <- m[5]
    }
```

Ορίζουμε τους γωνιακούς περιορισμούς.

```
    #### CR Constraints
    for(k in 1:3) {
      alpha[1,k] <- 0.0
      beta[1,k] <- 0.0
    }
       ab[1,1] <- 0.0
    ab[1,2] <- 0.0
    for (k in 2:LB){
    ab[k,1] <- 0.0
    }
```

Ορίζουμε τις ανεξάρτητες priors χαμηλής πληροφορίας.

```
      # priors
    #
    for (k in 1:5){
      m[k]~dnorm( 0.0, 1.0E-04)
      tau[k] ~dgamma( 0.01, 0.01)  # precission
      s[k] <- sqrt(1/tau[k])
    }
```

```
    for (k in 1:3){
      alpha[2,k]~dnorm(0.0, 1.0E-04)
      for (b in 2:LB){
        beta[b,k]~dnorm(0.0, 1.0E-04)
      }
    }


    for (a in 2:LA){
        for (b in 2:LB){
              ab[b,a]~dnorm( 0.0, 1.0E-04)
    }}
```

Επίσης ορίζουμε τις συναρτήσεις κεντρικότητας του 0 (p0) για τις παραμέτρους alpha[,], beta[,], ab[,].

```
     #p0
    for (k in 1:3) {
      for (a in 1:LA){
        p.alpha[a,k] <- step(alpha[a,k])
      }
      for (b in 2:LB){
        p.beta[b,k] <- step(beta[b,k])
      }
    }
      for (a in 2:LA){ for (b in 2:LB) {
        p.ab[b,a] <- step(ab[b,a])
      }}


}
```

Τέλος, ορίζουμε τις αρχικές τιμές των “effective” παραμέτρων .

```
INITS
list( m=c( 1.0, 1.0, 1.0, 1.0, 1.0 ), alpha=structure(.Data=c(NA,
NA, NA, 0, 0, 0 ), .Dim=c(2,3)), tau=c( 1.0, 1.0, 1.0, 1.0, 1.0 ),
beta=structure(.Data=c(NA, NA, NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0), .Dim=c(5,3)), ab=structure(.Data=c(NA,NA, NA, 0, NA, 0, NA, 0,
NA, 0), .Dim=c(5,2)) )
```

Ερμηνευτικά ή προβλεπτικά μοντέλα

Μετατρέπουμε τα δεδομένα σε μορφή λίστας ώστε να τα εισάγουμε στο OpenBUGS.

```
dput(sm)
```

Συνεχίζουμε γράφοντας τον κώδικα του μοντέλου στο "**OpenBUGS**".

<u>Ξεκινούμε με το μοντέλο των κατηγορικών συμμεταβλητών και των αλληλεπιδράσεων των μεταβλητών car, distance, sex.</u>

Θα προχωρήσουμε στην κατασκευή των ψευδομεταβλητών (dummy variables). Η μέθοδος που θα χρησιμοποιήσουμε είναι αυτή του μηδενικού αθροίσματος (STZ) η οποία φαίνεται να έχει κάποια πλεονεκτήματα και θα μας δώσει και ένα συνολικό μέσο - προφίλ τυπικού καταναλωτή.

```
model{
  # Dummies
  #
  # Dummies for car , sex with 2 levels
  for ( i  in 1 : n ){
    # stz parametrization
    Dcar [i] <- equals( car[i], 1 )- equals( car[i], 0 )
    Dgender [i] <- equals( sex[i], 1 )- equals( sex[i], 0 )
  }

  # Dummies for distance with 4 levels
  for ( i  in 1 : n ){ for ( j  in 1 : 4 ){
         # stz parametrization
    Ddistance [i,j] ←equals(distance[i],j)-
equals( distance[ i ], 1)
  }}

  # Dummies for prices, Qfresh, Qpack, Queues with 5 levels
  for ( i  in 1 : n ){ for ( j  in 1 : 5 ){
    # stz parametrization
    DQfresh [i,j] <- equals( Qfresh[i], j ) - equals( Qfresh[i], 1 )
    DQpack [i,j] <- equals( Qpack[i], j ) - equals( Qpack[i], 1 )
    DQueues [i,j] <- equals( Queues[i], j ) - equals( Queues[i], 1 )
    Dprices [i,j] <- equals( prices[i], j ) - equals( prices[i], 1 )
  }}

  # Dummies for shop with 10 levels
  for ( i  in 1 : n ){ for ( j  in 1 : 10 ){
     # stz parametrization
     Dshop [i,j] <- equals( shop[i], j ) - equals( shop[i], 1 )
  }}

    # Dummies for interaction term
  for ( i  in 1 : n ){ for ( j  in 1 : 4 ){
```

```
    Dcar.distance [ i , j ]  <-  Dcar[i] * Ddistance[i,j]
    Dgender.distance [ i , j ]  <-  Dgender[i] * Ddistance[i,j]
 }}
```

Φτιάχνουμε τη συνάρτηση πιθανοφάνειας του μοντέλου μας (likelihood function). Σε αυτή ενσωματώνουμε και τα μοντέλα των επεξηγηματικών μεταβλητών που χρησιμοποιούμε για τον υπολογισμό των ελλιπών τιμών. Η μεταβλητή car, αναμένουμε να εξαρτάται από τη distance.

```
 for ( i  in 1 : n ){

   # model's likelihood
   mu [i] <- m + alpha1 [2] * Dcar [i]
       + alpha2 [2] * Dgender [i]
       + inprod(alpha3 [2:4] , Ddistance [i, 2:4] )
       + inprod(alpha4 [2:5] , DQfresh [i, 2:5] )
       + inprod(alpha5 [2:5] , DQpack [i, 2:5] )
       + inprod(alpha6 [2:5] , DQueues [i, 2:5] )
       + inprod(alpha7 [2:5] , Dprices [i, 2:5] )
       + inprod(alpha8 [2:10] , Dshop [i, 2:10] )
       + alpha9 [ 2 ] * Dgender[i] * Dcar[i]
       + inprod(alpha10 [ 2:4 ] , Dcar.distance [ i , 2:4 ]  )
       + inprod(alpha11 [ 2:4 ] , Dgender.distance [ i ,2:4 ]  )

   spen_yr [i] ~ dnorm(mu [i] , tau )
   #
   # NAs estimation
   # model for Xs
   car [ i ] ~ dbern(p1[i]  )
   logit(p1[i] )<- mc + b[1]*Ddistance[i,2] + b[2]*Ddistance[i,3]
                                            + b[3]*Ddistance[i,4]
   sex [i] ~ dbern ( p2 )
   #
   distance [i] ~ dcat( pd[1:4 ] )
   prices [i] ~ dcat( pp[1:5] )
   Qfresh [i] ~ dcat( pQf[1:5] )
   Qpack [i] ~ dcat( pQp[1:5] )
   Queues [i] ~ dcat( pQu[1:5] )
   #
   shop [ i ]  ~  dcat ( ps [ 1:10 ] )

}
```

Γράφουμε τους περιορισμούς μηδενικού αθροίσματος των ψευδο-μεταβλητών.

```
 alpha1  [ 1 ] <- - alpha1 [ 2 ]
 alpha2  [ 1 ] <- - alpha2 [ 2 ]
 alpha3  [ 1 ] <- - sum( alpha3 [ 2 : 4 ]  )
 alpha4  [ 1 ] <- - sum( alpha4 [ 2 : 5 ]  )
 alpha5  [ 1 ] <- - sum( alpha5 [ 2 : 5 ]  )
 alpha6  [ 1 ] <- - sum( alpha6 [ 2 : 5 ]  )
 alpha7  [ 1 ] <- - sum( alpha7 [ 2 : 5 ]  )
 alpha8  [ 1 ] <- - sum( alpha8 [ 2 : 10 ]  )
```

```
alpha9  [ 1 ] <- - alpha9[2]
alpha10  [ 1 ] <- - sum( alpha10 [ 2 : 4 ]  )
alpha11  [ 1 ] <- - sum( alpha11 [ 2 : 4 ]  )
```

Συνεχίζουμε ορίζοντας τις prior κατανομές.

```
  # priors
  #model priors
  m ~ dnorm ( 0.0 , 1.0E-04 )
  tau ~ dgamma (0.01 , 0.01 )
  s <- sqrt( 1/tau )

  alpha1 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )
  alpha2 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )
  alpha9 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )

  for ( i in 2 : 4 ){ alpha3 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                      alpha10 [ i ]  ~ dnorm( 0.0 , 1.0E-04 )
                      alpha11 [ i ] ~ dnorm(0.0 , 1.0E-04 ) }

  for ( i in 2 : 5 ){ alpha4 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                      alpha5 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                      alpha6 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                      alpha7 [ i ] ~ dnorm(0.0 , 1.0E-04 )  }
  #for model Xs
  mc~dnorm(0,0.001)
  for(i in 1:3){ b[i]~dnorm(0,0.001)}
  p2~dbeta(1,1)
  #
  #
  for (k in 1:18){a[k]<-1 }
  #
  pd[1:4]~ddirch(a[ 1:4 ])
  pQf[1:5]~ddirch(a[ 1:5 ])
  pQp[1:5]~ddirch(a[ 1:5 ])
  pp[1:5]~ddirch(a[ 1:5 ])
  pQu[1:5]~ddirch(a[ 1:5 ])
  ps[1:10]~ddirch(a[ ] )

}
```

Τέλος, ορίζουμε την μπεύζιανή εκδοχή του συντελεστή προσδιορισμού $\mathbf{R^{2,}}$ και τις συναρτήσεις **p0**.

```
#pseudo-R
sy2<-pow(sd(spen_yr[]),2)
s2<-1/tau
R2B<-1-s2/sy2
```

```
#p0
p.alpha1[2 ] <- step(alpha1[2] )
p.alpha2[2 ] <- step(alpha2[2] )
p.alpha9[2 ] <- step(alpha2[2] )
```

```
  for (k in 1:4 ){p.alpha3[k] <- step(alpha3[k])
                  p.alpha10[k] <- step(alpha3[k])
                  p.alpha11[k] <- step(alpha3[k])
}
  for (k in 1:5 ){p.alpha4[k] <- step(alpha4[k])
                  p.alpha5[k] <- step(alpha5[k])
                  p.alpha6[k] <- step(alpha6[k])
                  p.alpha7[k] <- step(alpha7[k])}
  for (k in 1:10 ){p.alpha8[k] <- step(alpha8[k])}
```

Σε αυτό το σημείο θα ορίσουμε τις αρχικές τιμές των παραμέτρων.

Για τις αρχικές τιμές των παραμέτρων των κατηγορικών μεταβλητών (p1, p2, pd, pp, pQf, pQp, pQu, ps) θα χρησιμοποιήσουμε τον τύπο $p_i = 1 / k$, όπου k το πλήθος των επιπέδων της κάθε μεταβλητής. Ουσιαστικά χρησιμοποιούμε την αναμενόμενη τιμή της μη πληροφοριακής (φλατ) κατανομής Βήτα και Dirichlet (πολυμεταβλητή Βήτα).

```
#INITS 1

list(
m = 1.0 ,  tau = 1.0 ,
alpha1 = c( NA, 0 ), alpha2 = c( NA , 0 ), alpha9 = c( NA , 0 )
alpha3 = c( NA , 0 , 0 , 0 ), alpha4 = c(NA , 0 , 0 , 0 , 0 ) ,
alpha5 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha6 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha7 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha8 = c( NA , 0 , 0 , 0 , 0  , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0
           , 0 , 0 , 0 , 0  ),
alpha10 = c( NA , 0 , 0 , 0 ), alpha11 = c( NA , 0 , 0 , 0 )
mc = 1.0, b = c( 0, 0, 0 ) , p2 = 0.5 ,
pd = c( 0.25 , 0.25 , 0.25 , 0.25 ) ,
pQf = c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 ) ,
pQp = c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 ) ,
pp = c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 ),
pQu = c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 ),
ps = c( 0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1 ))
```

Οι αρχικές τιμές παίζουν καθοριστικό ρόλο στη σύγκλιση και στον καλύτερο υπολογισμό των τιμών των posterior κατανομών. Θα χρησιμοποιήσουμε και έναν άλλο τρόπο για τον καθορισμό των αρχικών τιμών των p_i , αυτόν της εκτίμησης των δειγματικών p_i ($p_i = n_i / n$, όπου n_i ο αριθμός "επιτυχιών" του i επιπέδου και n το μέγεθος του δείγματος της κάθε μεταβλητής.

Για τον υπολογισμό αυτών των τιμών επιστρέφουμε στην **R** και εκτελούμε τις ακόλουθες εντολές.

```
#p1
p1<-sum(na.omit(sm$car==1))/length(na.omit(sm$car))
#p2
p2<-sum(na.omit(sm$sex==1))/length(na.omit(sm$sex))
#pd[]
```

```
pd<-NULL
for (i in 1:4) {
 pd[i]<-sum(na.omit(sm$distance==i))/length(na.omit(sm$distance))
}
#pp[], pQf[], pQp[], pQu[]
pp<-NULL
pQf<-NULL
pQp<-NULL
pQu<-NULL
for (i in 1:5) {
  pp[i]<-sum(na.omit(sm$prices==i))/length(na.omit(sm$prices))
  pQf[i]<-sum(na.omit(sm$Qfresh==i))/length(na.omit(sm$Qfresh))
  pQp[i]<-sum(na.omit(sm$Qpack==i))/length(na.omit(sm$Qpack))
  pQu[i]<-sum(na.omit(sm$Queues==i))/length(na.omit(sm$Queues))
}
#ps
ps<-NULL
for (i in 1:10) {
  ps[i]<-sum(na.omit(sm$shop==i))/length(na.omit(sm$shop))
}

l1<-list(p1=p1,p2=p2,pd=pd,pp=pp,pQf=pQf,pQp=pQp,pQu=pQu,ps=ps)
dput(l1)
```

Έχουμε λοιπόν τις παρακάτω αρχικές τιμές.

```
#INITS 2

list(
m = 1.0 ,  tau = 1.0 ,
alpha1 = c( NA, 0 ), alpha2 = c( NA , 0 ), alpha9 = c( NA , 0 )
alpha3 = c( NA , 0 , 0 , 0 ), alpha4 = c(NA , 0 , 0 , 0 , 0 ) ,
alpha5 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha6 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha7 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha8 = c( NA , 0 , 0 , 0 , 0  , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0
           , 0 , 0 , 0 , 0  ),
alpha10 = c( NA , 0 , 0 , 0 ), alpha11 = c( NA , 0 , 0 , 0 )
mc = 1.0, b = c( 0, 0, 0 ), p2 = 0.228155339805825,
pd = c( 0.231017770597738 , 0.701130856219709 , 0.0516962843295638 ,
        0.0161550888529887 ) ,
pp = c( 0.0311986863711002 , 0.0213464696223317, 0.09688013136289,
        0.147783251231527 , 0.702791461412151 ) ,
pQf = c( 0.0296540362438221 ,  0.0230642504118616,
         0.0774299835255354 , 0.130148270181219,
         0.739703459637562 ) ,
pQp = c( 0.0264026402640264 , 0.0412541254125413, 0.127062706270627,
         0.217821782178218, 0.587458745874587 ),
pQu = c(0.0282861896838602 , 0.0382695507487521 , 0.131447587354409,
        0.176372712146423 , 0.625623960066556),
ps  = c(0.0547504025764895, 0.035426731078905, 0.0402576489533011,
        0.154589371980676, 0.035426731078905, 0.0450885668276973,
        0.0402576489533011, 0.0901771336553945, 0.0692431561996779,
```

```
        0.434782608695652)
```

Τα δεδομένα μας και ο κώδικας προσαρμογής του μοντέλου στην R μέσω “R2WinBUGS” είναι διαθέσιμα στα παρακάτω αρχεία.

¸ããñáöï êåéìÝíïõ

¸ããñáöï êåéìÝíïõ

<u>Μοντέλο κατηγορικών συμμεταβλητών και των αλληλεπιδράσεων με τη μεταβλητή shop.</u>

Κατασκευή ψευδομεταβλητών και interaction μεταβλητών.

```
model{

  # Dummies#
  #stz parametrization
  # Dummies for car , sex with 2 levels
  for ( i  in 1 : n ){
     Dcar [i] <- equals( car[i] , 1 )- equals( car [i] , 0  )
     Dgender [i] <- equals(sex[i] , 1 )- equals( sex[i] , 0  )

  # Dummies for distance with 4 levels
    for ( j  in 1 : 4 ){
      Ddistance [i,j] <- equals(distance[i],j)-
equals(distance[i],1)
  }
  # Dummies for prices, Qfresh, Qpack, Queues with 5 levels
    for ( j  in 1 : 5 ){
    DQfresh [i ,j]  <- equals( Qfresh [i],j)-
equals( Qfresh [i],1)
    DQpack [i ,j]  <- equals( Qpack [i] ,j)-equals(Qpack [i],1 )
    DQueues [i,j] <- equals( Queues[i],j)-equals( Queues [i],1)
    Dprices [i ,j]<-equals( prices [i], j)-equals(prices[i],1 )
  }
```

```
# Dummies for shop with 10 levels
 for ( j  in 1 : 10 ){
   Dshop [i,j]  <- equals(shop [i],j)- equals(shop[i],1)
}

# Dummies for interaction term
 for ( j  in 1 : 10 ){ for ( k in 1:4 ) {
    Dshop.Queues [i, j, k]  <-  Dshop[i,j] * DQueues[i,k]
    Dshop.Qfresh [i, j, k]  <-  Dshop[i,j] * DQfresh[i,k]
    Dshop.Qpack [i, j, k]  <-  Dshop[i,j] * DQpack[i,k]
         Dshop.prices [i, j, k]  <-  Dshop[i,j] * Dprices[i,k]
 }}
}
```

Για τη συνάρτηση πιθανοφάνειας του μοντέλου (likelihood function), έχουμε:

```
for ( i  in 1 : n ){
  # model's likelihood
  mu [ i ] <- m + alpha1 [2] * Dcar [i]
         + alpha2 [2] * Dgender [i]
              + inprod(alpha3[2:4], Ddistance[i,2:4])
         + inprod(alpha4[2:5], DQfresh[i,2:5])
         + inprod(alpha5[2:5], DQpack[i,2:5])
         + inprod(alpha6[2:5], Dqueues[i,2:5])
         + inprod(alpha7[2:5], Dprices[i,2:5])
         + inprod(alpha8[2:10], Dshop[i,2:10])
         + inprod(alpha9[2:10,2], Dshop.Qfresh[i,2:10,2])
         + inprod(alpha9[2:10,3], Dshop.Qfresh[i,2:10,3])
           + inprod(alpha9[2:10,4], Dshop.Qfresh[i,2:10,4])
                + inprod(alpha10[2:10,2], Dshop.Qpack[i,2:10,2])
                + inprod(alpha10[2:10,3], Dshop.Qpack[i,2:10,3])
                + inprod(alpha10[2:10,4], Dshop.Qpack[i,2:10,4])
         + inprod(alpha11[2:10,2], Dshop.Queues[i,2:10,2])
         + inprod(alpha11[2:10,3], Dshop.Queues[i,2:10,3])
         + inprod(alpha11[2:10,4], Dshop.Queues[i,2:10,4])
         + inprod(alpha12[2:10,2], Dshop.prices[i,2:10,2])
         + inprod(alpha12[2:10,3], Dshop.prices[i,2:10,3])
           + inprod(alpha12[2:10,4], Dshop.prices[i,2:10,4])


  spen_yr [ i ] ~ dnorm(mu [ i ] , tau )
  #
  # NAs estimation
  # model for Xs
  car [ i ] ~ dbern(p1[i]  )
  logit(p1[i] )<- mc + b[1]*Ddistance[i,2] +     `
b[1]*Ddistance[i,3] + b[1]*Ddistance[i,4]
  sex [ i ] ~ dbern (p2  )
  #
  distance [ i ] ~ dcat( pd[1:4 ] )
  prices [ i ] ~ dcat( pp[1:5] )
```

```
    Qfresh [ i ] ~ dcat( pQf[1:5] )
    Qpack [ i ] ~ dcat( pQp[1:5] )
    Queues [ i ] ~ dcat( pQu[1:5] )
    #
    shop [ i ] ~ dcat(ps[1:10])

  }
```

Γράφουμε τους περιορισμούς μηδενικού αθροίσματος των ψευδο-μεταβλητών.

```
  ###  stz constraints
  alpha1  [ 1 ] <- - alpha1 [ 2 ]
  alpha2  [ 1 ] <- - alpha2 [ 2 ]
  alpha3  [ 1 ] <- - sum( alpha3 [ 2 : 4 ]  )
  alpha4  [ 1 ] <- - sum( alpha4 [ 2 : 5 ]  )
  alpha5  [ 1 ] <- - sum( alpha5 [ 2 : 5 ]  )
  alpha6  [ 1 ] <- - sum( alpha6 [ 2 : 5 ]  )
  alpha7  [ 1 ] <- - sum( alpha7 [ 2 : 5 ]  )
  alpha8  [ 1 ] <- - sum( alpha8 [ 2 : 10 ]  )
  for ( j in 1:4 ) {
  alpha9  [ 1, j ] <- - sum( alpha9 [ 2 : 10, j ]  )
  alpha10  [ 1, j ] <- - sum( alpha10 [ 2 : 10, j ]  )
  alpha11  [ 1, j ] <- - sum( alpha11 [ 2 : 10, j ]  )
  alpha12  [ 1, j ] <- - sum( alpha12 [ 2 : 10, j ]  )
  }
  for ( i in 2:10){
  alpha9  [ i, 1] <- - sum( alpha9 [ i, 2:4 ]  )
  alpha10  [ i, 1 ] <- - sum( alpha10 [ i, 2:4 ]  )
  alpha11  [ i, 1 ] <- - sum( alpha11 [ i, 2:4 ]  )
  alpha12  [ i, 1 ] <- - sum( alpha12 [ i, 2:4 ]  )
  }
```

Συνεχίζουμε ορίζοντας τις prior κατανομές.

```
  # priors
  #model priors
  m ~ dnorm ( 0.0 , 1.0E-04 )
  alpha1 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )
  alpha2 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )
  for ( i in 2 : 4 ){ alpha3 [ i ] ~ dnorm(0.0 , 1.0E-04 ) }
  for ( i in 2 : 5 ){ alpha4 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                      alpha5 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                        alpha6 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                      alpha7 [ i ] ~ dnorm(0.0 , 1.0E-04 ) }
  for ( i in 2 : 10 ){ alpha8 [ i ] ~ dnorm(0.0 , 1.0E-04 ) }
  for ( i in 2 : 10 ){ for ( j in 2:4 ){
                     alpha9  [ i, j ]  ~ dnorm(0.0 , 1.0E-04 )
                     alpha10  [ i, j ]  ~ dnorm(0.0 , 1.0E-04 )
                     alpha11  [ i, j ]  ~ dnorm(0.0 , 1.0E-04 )
                     alpha12  [ i, j ]  ~ dnorm(0.0 , 1.0E-04 ) }}

  tau ~ dgamma (0.01 , 0.01 )
  s <- sqrt( 1/tau )
```

```
   #for model Xs
 mc~dnorm(0,0.001)
 for(i in 1:3){ b[i]~dnorm(0,0.001)}
 p2~dbeta(1,1)
 #
 #
 for (k in 1:10){a[k]<-1 }
 pd[1:4]~ddirch(a[ 1:4 ])
 pQf[1:5]~ddirch(a[ 1:5 ])
 pQp[1:5]~ddirch(a[ 1:5 ])
 pp[1:5]~ddirch(a[ 1:5 ])
 pQu[1:5]~ddirch(a[ 1:5 ])
   ps[1:10]~ddirch(a[ ] )
```

Τέλος, ορίζουμε την μπεύζιανή εκδοχή του συντελεστή προσδιορισμού $\mathbf{R}^{2,}$ και τις συναρτήσεις **p0**.

```
 #pseudo-R
 sy2<-pow(sd(spen_yr[]),2)
 s2<-1/tau
 R2B<-1-s2/sy2
```

```
 #p0
 p.alpha1[2 ] <- step(alpha1[2] )
 p.alpha2[2 ] <- step(alpha2[2] )
 for (k in 2:4 ){p.alpha3[k] <- step(alpha3[k]) }
 for (k in 2:5 ){p.alpha4[k] <- step(alpha4[k])
           p.alpha5[k] <- step(alpha5[k])
               p.alpha6[k] <- step(alpha6[k])
               p.alpha7[k] <- step(alpha7[k]) }
 for (k in 2:10 ){p.alpha8[k] <- step(alpha8[k]) }
 for ( i in 2 : 10 ){ for ( j in 2:4 ){
                         p.alpha9[i,j]  <- step(alpha9[i,j])
                         p.alpha10[i,j]  <- step(alpha10[i,j])
                         p.alpha11[i,j]  <- step(alpha11[i,j])
                         p.alpha12[i,j]  <- step(alpha12[i,j]) }}
}
```

Οι αρχικές τιμές του μοντέλου θα είναι.

```
#INITS 1

list(
m = 1.0 ,  tau = 1.0 ,
alpha1 = c( NA, 0 ), alpha2 = c( NA , 0 ),
alpha3 = c( NA , 0 , 0 , 0 ), alpha4 = c(NA , 0 , 0 , 0 , 0 ) ,
alpha5 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha6 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha7 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha8 = c( NA , 0 , 0 , 0 , 0  , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0
          , 0 , 0 , 0 , 0  ),
```

```
alpha9 = structure(.Data=c(NA,NA,NA,NA, NA, 0, 0, 0, NA, 0, 0, 0,
NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA,
0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0), .Dim=c(10,4)) ,
alpha10 = structure(.Data=c(NA,NA,NA,NA, NA, 0, 0, 0, NA, 0, 0, 0,
NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA,
0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0), .Dim=c(10,4)),
alpha11 = structure(.Data=c(NA,NA,NA,NA, NA, 0, 0, 0, NA, 0, 0, 0,
NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA,
0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0), .Dim=c(10,4)),
alpha12 = structure(.Data=c(NA,NA,NA,NA, NA, 0, 0, 0, NA, 0, 0, 0,
NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA,
0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0), .Dim=c(10,4)),
mc = 1.0, b = c( 0, 0, 0 ) , p2 = 0.5 ,
pd = c( 0.25 , 0.25 , 0.25 , 0.25 ) ,
pQf = c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 ) ,
pQp = c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 ) ,
pp = c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 ),
pQu = c( 0.2 , 0.2 , 0.2 , 0.2 , 0.2 ),
ps = c( 0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1,  0.1 ))
```

```
#INITS 2

list(
m = 1.0 ,  tau = 1.0 ,
alpha1 = c( NA, 0 ), alpha2 = c( NA , 0 ), alpha9 = c( NA , 0 )
alpha3 = c( NA , 0 , 0 , 0 ), alpha4 = c(NA , 0 , 0 , 0 , 0 ) ,
alpha5 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha6 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha7 = c( NA , 0 , 0 , 0 , 0 ) ,
alpha8 = c( NA , 0 , 0 , 0 , 0  , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0
           , 0 , 0 , 0 , 0  ),
alpha9 = structure(.Data=c(NA,NA,NA,NA, NA, 0, 0, 0, NA, 0, 0, 0,
NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA,
0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0), .Dim=c(10,4)) ,
alpha10 = structure(.Data=c(NA,NA,NA,NA, NA, 0, 0, 0, NA, 0, 0, 0,
NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA,
0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0), .Dim=c(10,4)),
alpha11 = structure(.Data=c(NA,NA,NA,NA, NA, 0, 0, 0, NA, 0, 0, 0,
NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA,
0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0), .Dim=c(10,4)),
alpha12 = structure(.Data=c(NA,NA,NA,NA, NA, 0, 0, 0, NA, 0, 0, 0,
NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0, NA,
0, 0, 0, NA, 0, 0, 0, NA, 0, 0, 0), .Dim=c(10,4)),
mc = 1.0, b = c( 0, 0, 0 ), p2 = 0.228155339805825,
pd = c( 0.231017770597738 , 0.701130856219709 , 0.0516962843295638 ,
        0.0161550888529887 ) ,
pp = c( 0.0311986863711002 , 0.0213464696223317, 0.09688013136289,
        0.147783251231527 , 0.702791461412151 ) ,
pQf = c( 0.0296540362438221 , 0.0230642504118616,
         0.0774299835255354 , 0.130148270181219,
         0.739703459637562 ) ,
pQp = c( 0.0264026402640264 , 0.0412541254125413, 0.127062706270627,
         0.217821782178218, 0.587458745874587 ),
```

```
pQu = c(0.0282861896838602 , 0.0382695507487521 , 0.131447587354409,
        0.176372712146423 , 0.625623960066556),
ps  = c(0.0547504025764895, 0.035426731078905, 0.0402576489533011,
        0.154589371980676, 0.035426731078905, 0.0450885668276973,
        0.0402576489533011, 0.0901771336553945, 0.0692431561996779,
        0.434782608695652)
```

<u>Μοντέλο “random effects”.</u>

Το μοντέλο “random effects” που θα προσαρμόσουμε είναι αντίστοιχο με εκείνο των κυρίων επιδράσεων, με τη διαφορά ότι εισάγουμε μία “hyper – prior” για την διακύμανση των συντελεστών της μεταβλητής “shop”.

Οι ψευδομεταβήτές που χρησιμοποιούμε είναι οι ίδιες χωρίς τα interactions.

Επίσης, η πιθανοφάνεια είναι ίδια με εκείνη των κυρίων επιδράσεων.

```
for ( i  in 1 : n ){
  # model's likelihood
  mu [ i ] <- m + alpha1 [2] * Dcar [i]
         + alpha2 [2] * Dgender [i]
              + inprod(alpha3[2:4], Ddistance[i,2:4])
         + inprod(alpha4[2:5], DQfresh[i,2:5])
         + inprod(alpha5[2:5], DQpack[i,2:5])
         + inprod(alpha6[2:5], Dqueues[i,2:5])
         + inprod(alpha7[2:5], Dprices[i,2:5])
         + inprod(alpha8[2:10], Dshop[i,2:10])


  spen_yr [ i ] ~ dnorm(mu [ i ] , tau )
  #
  # NAs estimation
  # model for Xs
  car [ i ] ~ dbern(p1[i]  )
  logit(p1[i] )<- mc + b[1]*Ddistance[i,2] +     `
b[1]*Ddistance[i,3] + b[1]*Ddistance[i,4]
  sex [ i ] ~ dbern (p2  )
  #
  distance [ i ] ~ dcat( pd[1:4 ] )
  prices [ i ] ~ dcat( pp[1:5] )
  Qfresh [ i ] ~ dcat( pQf[1:5] )
  Qpack [ i ] ~ dcat( pQp[1:5] )
  Queues [ i ] ~ dcat( pQu[1:5] )
  #
  shop [ i ] ~ dcat(ps[1:10])

}
```

Οι priors του μοντέλου ορίζονται ως εξής.

```
# priors
#model priors
m ~ dnorm ( 0.0 , 1.0E-04 )
alpha1 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )
alpha2 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )
for ( i in 2 : 4 ){ alpha3 [ i ] ~ dnorm(0.0 , 1.0E-04 ) }
for ( i in 2 : 5 ){ alpha4 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                    alpha5 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                      alpha6 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                    alpha7 [ i ] ~ dnorm(0.0 , 1.0E-04 ) }
for ( i in 2 : 10 ){ alpha8 [ i ] ~ dnorm(0.0 , tau.shop ) }

tau ~ dgamma (0.01 , 0.01 )
  tau.shop ~ dgamma (0.01 , 0.01 )
s <- sqrt( 1/tau )
  s.hop <- sqrt( 1/tau.shop )

#for model Xs
mc~dnorm(0,0.001)
for(i in 1:3){ b[i]~dnorm(0,0.001)}
p2~dbeta(1,1)
#
#
for (k in 1:10){a[k]<-1 }
pd[1:4]~ddirch(a[ 1:4 ])
pQf[1:5]~ddirch(a[ 1:5 ])
pQp[1:5]~ddirch(a[ 1:5 ])
pp[1:5]~ddirch(a[ 1:5 ])
pQu[1:5]~ddirch(a[ 1:5 ])
  ps[1:10]~ddirch(a[ ] )
```

Κατά τον ίδιο τρόπο ορίζουμε τους περιορισμούς των alpha1[1], alpha2[1], alpha3[1] κλπ. και τις αρχικές τιμές των παραμέτρων.

<u>Μοντέλο κύριων επιδράσεων κατηγορικών συμμεταβλητών, χωρίς τη μεταβλητή "shop".</u>

Οι ψευδομεταβλητές (Dummies) του μοντέλου είναι οι ίδιες με τις προηγούμενες περιπτώσεις, χωρίς τις ψευδομεταβλητές που κατασκευάσαμε για τα interactions και τη μεταβλητή "shop".

Για τη συνάρτηση πιθανοφάνειας του μοντέλου (likelihood function), έχουμε:

```
for ( i  in 1 : n ){
  # model's likelihood
  mu [ i ] <- m + alpha1 [2] * Dcar [i]
         + alpha2 [2] * Dgender [i]
              + inprod(alpha3[2:4], Ddistance[i,2:4])
         + inprod(alpha4[2:5], DQfresh[i,2:5])
```

```
        + inprod(alpha5[2:5], DQpack[i,2:5])
        + inprod(alpha6[2:5], Dqueues[i,2:5])
        + inprod(alpha7[2:5], Dprices[i,2:5])

  spen_yr [ i ] ~ dnorm(mu [ i ] , tau )
  #
  # NAs estimation
  # model for Xs
  car [ i ] ~ dbern(p1[i]  )
  logit(p1[i] )<- mc + b[1]*Ddistance[i,2] +     `
b[1]*Ddistance[i,3] + b[1]*Ddistance[i,4]
  sex [ i ] ~ dbern (p2  )
  #
  distance [ i ] ~ dcat( pd[1:4 ] )
  prices [ i ] ~ dcat( pp[1:5] )
  Qfresh [ i ] ~ dcat( pQf[1:5] )
  Qpack [ i ] ~ dcat( pQp[1:5] )
  Queues [ i ] ~ dcat( pQu[1:5] )
  #
}
```

Συνεχίζουμε ορίζοντας τις prior κατανομές.

```
# priors
#model priors
m ~ dnorm ( 0.0 , 1.0E-04 )
alpha1 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )
alpha2 [ 2 ]  ~ dnorm( 0.0 , 1.0E-04 )
for ( i in 2 : 4 ){ alpha3 [ i ] ~ dnorm(0.0 , 1.0E-04 ) }
for ( i in 2 : 5 ){ alpha4 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                    alpha5 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                      alpha6 [ i ] ~ dnorm(0.0 , 1.0E-04 )
                    alpha7 [ i ] ~ dnorm(0.0 , 1.0E-04 ) }


tau ~ dgamma (0.01 , 0.01 )
s <- sqrt( 1/tau )
#for model Xs
mc~dnorm(0,0.001)
for(i in 1:3){ b[i]~dnorm(0,0.001)}
p2~dbeta(1,1)
#
#
for (k in 1:10){a[k]<-1 }
pd[1:4]~ddirch(a[ 1:4 ])
pQf[1:5]~ddirch(a[ 1:5 ])
pQp[1:5]~ddirch(a[ 1:5 ])
pp[1:5]~ddirch(a[ 1:5 ])
pQu[1:5]~ddirch(a[ 1:5 ])
```

Κατά τον ίδιο τρόπο ορίζουμε τους περιορισμούς μηδενικού αθροίσματος των ψευδο-μεταβλητών.

<u>Μοντέλο ποσοτικών και κατηγορικών συμμεταβλητών (ANCOVA)”.</u>

Οι ψευδομεταβλητές (Dummies) του μοντέλου, αφορούν μόνο τις μεταβλητές car, sex και ορίζονται ακριβώς με τον ίδιο τρόπο όπως προηγουμένως.

Για τη συνάρτηση πιθανοφάνειας του μοντέλου (likelihood function), έχουμε:

```
for ( i  in 1 : n ){
  # model's likelihood
  mu [ i ] <- m + alpha1 [2] * Dcar [i]
         + alpha2 [2] * Dgender [i]
              + alpha3 * (distance [i] – mean( distance [] ) )
         + alpha4 * (Qfresh [i] -  mean(Qfresh []) )
         + alpha5 * (Qpack [i] - mean(Qpack []) )
         + alpha6 * (Queues [i]  - mean(Queues []) )
         + alpha7 * (prices [i] -  mean(prices []) )

  spen_yr [ i ] ~ dnorm(mu [ i ] , tau )
  #
  # NAs estimation
  # model for Xs
  car [ i ] ~ dbern(p1[i]  )
  logit(p1[i] )<- mc + b[1]*Ddistance[i,2] +     `
b[1]*Ddistance[i,3] + b[1]*Ddistance[i,4]
  sex [ i ] ~ dbern (p2  )
  #
  distance [ i ] ~ dcat( pd[1:4 ] )
  prices [ i ] ~ dcat( pp[1:5] )
  Qfresh [ i ] ~ dcat( pQf[1:5] )
  Qpack [ i ] ~ dcat( pQp[1:5] )
  Queues [ i ] ~ dcat( pQu[1:5] )
  #
}
```

Οι prior κατανομές και αρχικές τιμές ορίζονται ακριβώς με τον ίδιο τρόπο.

Επιλογή μεταβλητών

Για τη διαδικασία στην R μέσω του πακέτου "BAS" για την επιλογή μεταβλητών, αυτό που θα χρειαστεί να κάνουμε πρώτα είναι να φτιάξουμε τις dummies μηδενικού αθροίσματος στην R. Για να το κάνουμε αυτό, θα χρησιμοποιήσουμε τις παρακάτω εντολές

```
Dcar <- matrix(0,dim(sm.new)[1],2)
Dsex <- matrix(0,dim(sm.new)[1],2)
Ddistance <- matrix(0,dim(sm.new)[1],4)
DQfresh <- matrix(0,dim(sm.new)[1],5)
DQpack <- matrix(0,dim(sm.new)[1],5)
DQueues <- matrix(0,dim(sm.new)[1],5)
Dprices <- matrix(0,dim(sm.new)[1],5)

for ( i  in 1 : dim(sm.new)[1] ){
             # Dummies for car , sex with 2 levels
             if ( sm.new$car[i]==0 ) { Dcar[i,2] <- -1 }
             if ( sm.new$car[i]==1 ) { Dcar[i,2] <- 1 }
             if ( sm.new$car[i]==0 ) { Dsex[i,2] <- -1 }
             if ( sm.new$car[i]==1 ) { Dsex[i,2] <- 1 }

  # Dummies for distance with 4 levels
  for ( j  in 2 : 4 ){
             if ( sm.new$distance[i]==1 ) { Ddistance[i,j] <- -1 }
             else if ( sm.new$distance[i]==j ) { Ddistance[i,j] <- 1
}
             else { Ddistance[i,j] <- 0 }
  }

  # Dummies for prices, Qfresh, Qpack, Queues with 5 levels
  for ( j  in 2 : 5 ){
             if ( sm.new$Qfresh[i]==1 ) { DQfresh[i,j] <- -1 }
             else if ( sm.new$Qfresh[i]==j ) { DQfresh[i,j] <- 1 }
             else { DQfresh[i,j] <- 0 }
             if ( sm.new$Qpack[i]==1 ) { DQpack[i,j] <- -1 }
             else if ( sm.new$Qpack[i]==j ) { DQpack[i,j] <- 1 }
             else { DQpack[i,j] <- 0 }
             if ( sm.new$Queues[i]==1 ) { DQueues[i,j] <- -1 }
             else if ( sm.new$Queues[i]==j ) { DQueues[i,j] <- 1 }
             else { DQueues[i,j] <- 0 }
             if ( sm.new$prices[i]==1 ) { Dprices[i,j] <- -1 }
             else if ( sm.new$prices[i]==j ) { Dprices[i,j] <- 1 }
             else { Dprices[i,j] <- 0 }
  }
}
```

Αποθηκεύουμε τα νέα δεδομένα μαζί με την μεταβλητή απόκρισης σε ένα νέο data frame.

```
sm.new$Dcar <- Dcar[,2]
sm.new$Dsex <- Dsex[,2]
sm.new$Dprices2 <- Dprices[,2]
sm.new$Dprices3 <- Dprices[,3]
sm.new$Dprices4 <- Dprices[,4]
sm.new$Dprices4 <- Dprices[,4]
sm.new$DQfresh2 <- DQfresh[,2]
sm.new$DQfresh3 <- DQfresh[,3]
sm.new$DQfresh4 <- DQfresh[,4]
sm.new$DQfresh5 <- DQfresh[,5]
sm.new$DQpack2 <- DQpack[,2]
sm.new$DQpack3 <- DQpack[,3]
sm.new$DQpack4 <- DQpack[,4]
sm.new$DQpack5 <- DQpack[i,5]
sm.new$DQueues2 <- DQueues[,2]
sm.new$DQueues3 <- DQueues[,3]
sm.new$DQueues4 <- DQueues[,4]
sm.new$DQueues5 <- DQueues[,5]
sm.new$Ddistance2 <- Ddistance[,2]
sm.new$Ddistance3 <- Ddistance[,3]
sm.new$Ddistance4 <- Ddistance[,4]

sm.n<- sm.new[ ,c(2,10:dim(sm.new)[2])]
```

Εκτελούμε τη διαδικασία του Variable Selection για τις διαφορετικές prior.

```
#hyper-g beta binomial
res1 <- bas.lm( spen_yr~., data=sm.n, prior='hyper-g', alpha=3 )
round(t(summary(res1, 10)[,-1]),2)   # top 10 montela
res1  # inclusion probabilities

#hyper-g uniform
res2 <- bas.lm( spen_yr~., data=sm.n, prior='hyper-g', alpha=3 ,
modelprior=uniform())
round(t(summary(res2, 10)[,-1]),2)   # top 10 montela
res2  # inclusion probabilities


#g-prior beta binomial
res3 <- bas.lm( spen_yr~., data=sm.n, prior='g-prior',
alpha=dim(sm.new)[2] )
round(t(summary(res3, 10)[,-1]),2)   # top 10 montela
res3  # inclusion probabilities

#g-prior uniform
res4 <- bas.lm( spen_yr~., data=sm.n, prior='g-prior',
alpha=dim(sm.new)[2] ,modelprior=uniform())
round(t(summary(res4, 10)[,-1]),2)   # top 10 montela
res4  # inclusion probabilities

#BIC beta binomial
```

```
res5 <- bas.lm( spen_yr~., data=sm.n, prior='BIC' )
round(t(summary(res5, 10)[,-1]),2)   # top 10 montela
res5  # inclusion probabilities

#BIC uniform
res6 <- bas.lm( spen_yr~., data=sm.n, prior='BIC',
modelprior=uniform())
round(t(summary(res6, 10)[,-1]),2)   # top 10 montela
res6  # inclusion probabilities
```

Εξάγουμε τα αποτελέσματα των “MAP” και “MP” για το καλύτερο μοντέλο.\

```
#coefficients
# highest probability model (map)
coef(res1, estimator='HPM', n.models=100)
# median probability model (MP)
coef(res1, estimator='MPM', n.models=100)
coef(res2, estimator='HPM', n.models=100)
coef(res2, estimator='MPM', n.models=100)
coef(res3, estimator='HPM', n.models=100)
coef(res3, estimator='MPM', n.models=100)
coef(res4, estimator='HPM', n.models=100)
coef(res4, estimator='MPM', n.models=100)
coef(res5, estimator='HPM', n.models=100)
coef(res5, estimator='MPM', n.models=100)
coef(res6, estimator='HPM', n.models=100)
coef(res6, estimator='MPM', n.models=100)



#shrinkage parameter for hyper-g prior
#
# BMA estimate of w
sum(res1$postprobs*res1$shrinkage)
sum(res2$postprobs*res2$shrinkage)

#
# BMA estimate of g
# (does not directly corresponds to the mean because
#  it is based on transformation of the mean of w)
w1<-res1$shrinkage
w1[w1==1] <- 1-exp(-10)
g1<-w1/(1-w1)
sum(res1$postprobs*g)

w2<-res1$shrinkage
w2[w2==1] <- 1-exp(-10)
g1<-w2/(1-w2)
sum(res2$postprobs*g)
```

Η ίδια διαδικασία θα ακολουθηθεί μέσω του OpenBUGS.

Για την **hyper-g prior** με **Βήτα Διωνυμική** κατανομή εφαρμόζουμε το παρακάτω μοντέλο. Τα prop.mean.beta και prop.sd.beta τα έχουμε εκτιμήσει από τις posterior κατανομές του μοντέλου που έχουμε τρέξει ήδη.

Αρχικά, κατασκευάζουμε τις dummies που θα εισαχθούν στη likelihood του μοντέλου μας.

```
model{

  # Dummies#
  #stz parametrization
  # Dummies for car, sex with 2 levels
  for (i in 1:n){
     Z[i,1] <- equals(car[i], 1)- equals(car[i], 0)
     Z[i,2] <- equals(sex[i], 1)- equals(sex[i], 0)
  }

  # Dummies for distance with 4 levels
  for (i in 1:n){ for (j in 2:4){
     Z[i, j+1] <- equals(distance[i], j)-equals(distance[i], 1)
  }}

  # Dummies for prices, Qfresh, Qpack, Queues with 5 levels
  for (i in 1:n){ for (j in 2:5){
    Z[i, j+4]  <- equals(Qfresh [i], j)- equals(Qfresh [i], 1)
    Z[i, j+8]  <- equals(Qpack [i], j)- equals(Qpack [i], 1)
    Z[i, j+12]  <- equals(Queues [i], j)- equals(Queues [i], 1)
    Z[i, j+16]  <- equals(prices [i], j)- equals(prices [i], 1)
  }}
```

Υπολογίζουμε τον ανάστροφο X πίνακα, των επεξηγηματικών μεταβλητών.

```
# calculating the xtx matrix
for (j in 1:p){ for (k in 1:p ) {
      XTX[j,k] <- inprod( Z[1:n,j], Z[1:n,k] )
 }}
```

Στην πορεία, ορίζουμε την likelihood του μοντέλου.

```
  #
  for (j in 1:p){  gb[j] <- B[j]*g[j] }
  for ( i  in 1 : n ){
    # model's likelihood
    mu [ i ] <- alpha + inprod( Z[i,], gb[] )
    spen_yr [ i ] ~ dnorm(mu [ i ] , tau )
    #
    # NAs estimation
    # model for Xs
    car [ i ] ~ dbern(p1[i]  )
    logit(p1[i] )<- mc + b[1]*Z[i,3] + b[1]*Z[i,4] + b[1]*Z[i,5]
    sex [ i ] ~ dbern (p2  )
```

```
    distance [ i ] ~ dcat( pd[1:4 ] )
    prices [ i ] ~ dcat( pp[1:5] )
    Qfresh [ i ] ~ dcat( pQf[1:5] )
    Qpack [ i ] ~ dcat( pQp[1:5] )
    Queues [ i ] ~ dcat( pQu[1:5] )
    #

  }
```

Ορίζουμε τις **prior** του μοντέλου.

```
#   priors
     alpha ~ dflat()
#      Beta Binomial on model space
    for (j in 1:p) { g[j]~dbern(prob)  }
      prob ~dbeta(1,1)

    B[1:p] ~ dmnorm( mean.beta[1:p], T[ 1:p, 1:p] )
    logtau ~ dflat()    # improper prior
    tau <- exp( logtau )

   #tau~dgamma( 0.01, 0.01)

    for (j in 1:p ){ mean.beta[j] <-  (1-g[j])*prop.mean.beta[j] }
    for (j in 1:p ){ for (k in 1:p ){
                 T[j,k] < - g[j]*g[k]*tau*XTX[j,k]/G + ( 1-
g[j]*g[k])*equals(j,k)*pow(prop.sd.beta[k],-2)
    }}

     for (j in 1:p){ mindex[j] <- pow(2,j)}    # index needed for the
calculation of model
     model <- 1 + inprod( g[], mindex[] )      # model
indicator/code

       # calculate the probabilities of best models
     pmodel[1] <- equals( model, 1+pow(2,4)+pow(2,5))
     pmodel[2] <- equals( model, 1+pow(2,4)+pow(2,5)+pow(2,12))

     # hyper-prior
     a <- 3
     bw <- a/2-1
     w ~ dbeta( 1, bw )
     G <- w/(1-w)

  #for model Xs
  mc~dnorm(0,0.001)
  for(i in 1:3){ b[i]~dnorm(0,0.001)}
  p2~dbeta(1,1)
  #
  #
  as[ 1 ]<-1
  as[ 2 ]<-1
  as[ 3 ]<-1
  as[ 4 ]<-1
```

```
    as[ 5 ]<-1
  pd[1:4]~ddirch(as[ 1:4 ])
  pQf[1:5]~ddirch(as[ 1:5 ])
  pQp[1:5]~ddirch(as[ 1:5 ])
  pp[1:5]~ddirch(as[ 1:5 ])
  pQu[1:5]~ddirch(as[ 1:5 ])

}
```

Τέλος, ορίζουμε τις αρχικές τιμές.

```
list( w=0.7, alpha=0.6,
B=c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), logtau=0,
g=c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), prob=0.5, b=c(0, 0,
0 ), p2=0.5, pd=c(0.25,0.25,0.25,0.25), pQf=c(0.2,0.2,0.2,0.2,0.2),
pQp=c(0.2,0.2,0.2,0.2,0.2), pp=c(0.2,0.2,0.2,0.2,0.2),
pQu=c(0.2,0.2,0.2,0.2,0.2))
```

Για την **hyper-g prior** με **Ομοιόμορφη** κατανομή το μόνο που αλλάζουμε είναι η prior για τα g[].

```
#   priors
     alpha ~ dflat()
#     Uniform on model space
       for (j in 1:p) { g[j]~dbern(0.5)  }
```

Τώρα, για την **g prior** με **Βήτα-Διωνυμική** κατανομή εφαρμόζουμε το παρακάτω μοντέλο με την αντίστοιχη πιθανοφάνεια, προσθέτοντας και ένα ακόμη μοναδιαίο διάνυσμα στον πίνακα των επεξηγηματικών μεταβλητών για τη σταθερά (δε διαχωρίζουμε τη σταθερά από τις υπόλοιπες παραμέτρους).

```
  for ( i  in 1 : n ){
   Z[i , 22 ] <-1 #stathera
  }  #stathera

# calculating the xtx matrix
for (j in 1:(p+1)){ for (k in 1:(p+1) ) {
      XTX[j,k] <- inprod( Z[1:n,j], Z[1:n,k] )
 }}
  #
  for (j in 1:(p+1)){  gb[j] <- B[j]*g[j] }
  for ( i  in 1 : n ){
    # model's likelihood
    mu [ i ] <- inprod( Z[i,], gb[] )
    spen_yr [ i ] ~ dnorm(mu [ i ] , tau )
```

Οι priors θα ορίζονται όπως στη συνέχεια.

```
#   priors
#     Beta Binomial on model space
       for (j in 1:(p+1)) { g[j]~dbern(prob)  }
        prob ~dbeta(1,1)


    B[1:(p+1)] ~ dmnorm( mean.beta[1:(p+1)], T[ 1:(p+1), 1:(p+1)] )
      tau~dgamma( 0.001, 0.001)

    for (j in 1:(p+1) ){ mean.beta[j] <-  (1-g[j])*prop.mean.beta[j]
}
    for (j in 1:(p+1)){ for (k in 1:(p+1) ){
                 T[j,k] < - g[j]*g[k]*tau*XTX[j,k]/n + ( 1-
g[j]*g[k])*equals(j,k)*pow(prop.sd.beta[k],-2)
    }}

     for (j in 1:p){ mindex[j] <- pow(2,j)}    # index needed for the
calculation of model
     model <- 1 + inprod( g[], mindex[] )      # model
indicator/code
```

Η αρχική τιμή της tau θα οριστεί ίση με 1.

Η **g prior** με **Ομοιόμορφη** κατανομή διαφέρει μόνο ως προς τον τρόπο που ορίζουμε την prior. Δεν ορίζουμε prior στη παράμετρο της κατανομής Bernoulli που ακολουθούν τα g[].

```
#   priors
#     Beta Binomial on model space
       for (j in 1:(p+1)) { g[j]~dbern(0.5)  }
```

Συνεχίζουμε με την **Unit Information empirical prior**. Η πιο απλή εκδοχή των priors, βασισμένη εξ' ολοκλήρου στην δειγματική posterior κατανομή. Το αν θα επιλέξουμε τη Βήτα-Διωνυμική ή την Ομοιόμορφη κατανομή μπορεί να γίνει μετατρέποντας τα σχόλια σε εντολές

```
for (j in 1:p){  gb[j] <- beta[j]*gamma[j] }
for (i in 1:n){
      spen_yr[i] ~ dnorm( mu[i], tau )
     mu[i] <- gamma0 * beta0 + inprod( Z[i,], gb[] )
```

Οι priors θα ορίζονται ως εξής.

```
#   priors
#
#     Constant is included always in the model
#     gamma0 ← 1
     gamma0 ~ dbern( 0.5 )
#
#     Uniform prior on models
     for (j in 1:p){ gamma[j]~dbern(0.5) }
#
#     Beta binomial on models
#     for (j in 1:p){ gamma[j]~dbern(p) }
#     p~dbeta(1,1)

#pseydo-prior
     beta0 ~ dnorm( mb0, taub0)
     mb0 <-   prop.mean.beta0
     taub0 <- (gamma0/n + (1-gamma0) )/pow(prop.sd.beta0 ,2)
     for (j in 1:p){
      beta[j] ~ dnorm( mb[j], taub[j])
      mb[j] <- prop.mean.beta[j]
      taub[j] <- 1/ (n* pow(prop.sd.beta[j] ,2))
     }

     tau~dgamma( 0.001, 0.001)
```

Πίνακες

Ανάλυση Διακύμανσης (ANOVA)

Πίνακας 1 "Πίνακας διαγνωστικού ελέγχου σύγκλισης του Geweke"

```
GEWEKE CONVERGENCE DIAGNOSTIC (Z-score)
=======================================

Iterations used = 2001:6000
Thinning interval = 1
Sample size per chain = 4000

$`C:/Users/part/Desktop/anova_all_models.out.txt`

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

 ab[2,2]  ab[3,2]  ab[4,2]  ab[5,2] alpha[2]  beta[2]  beta[3]
 1.66362  0.49446 -0.69687 -0.28729 -1.62730  0.20645 -0.93837

 beta[4]     beta[5]        m        s         tau
 -1.34285    -0.80914  -0.03325  1.58826    -1.41987
```

Ο έλεγχος σύγκλισης του Gewek, όπως φαίνεται στον *Πίνακα 1*, μας δίνει τα z-scores των παραμέτρων τα οποία δεν πρέπει να είναι στα άκρα της τυποποιημένης κανονικής κατανομής αν η σύγκλιση του αλγορίθμου είναι επιτυχής. Οι τιμές των z-scores είναι μικρότερες του 1.66 (ab[2,2]) και μεγαλύτερες του -1.63 (alpha[2]). Και οι δύο αυτές τιμές αντιστοιχούν σε ποσοστιαία σημεία της κατανομής αρκετά μικρότερα του 95% και μεγαλύτερα του 5% αντίστοιχα συνεπώς δε βρίσκονται στα άκρα της κατανομής. Με μεγάλη πιθανότητα λοιπόν θα λέγαμε ότι ο αλγόριθμός μας συγκλίνει.

Πίνακας 2 "Πίνακας διαγνωστικού ελέγχου σύγκλισης 'Raftery and Lewis'."

```
RAFTERY AND LEWIS CONVERGENCE DIAGNOSTIC
========================================

Iterations used = 2001:6000
Thinning interval = 1
Sample size per chain = 4000

$`C:/Users/part/Desktop/anova_all_models.out.txt`

Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

          Burn-in  Total Lower bound  Dependence
          (M)      (N)   (Nmin)       factor (I)
 ab[2,2]  2        3866  3746         1.030
 ab[3,2]  2        3668  3746         0.979
 ab[4,2]  2        3866  3746         1.030
 ab[5,2]  2        3790  3746         1.010
 alpha[2] 2        3711  3746         0.991
 beta[2]  2        3711  3746         0.991
 beta[3]  2        3787  3746         1.010
 beta[4]  2        3635  3746         0.970
 beta[5]  3        4406  3746         1.180
 m        3        4357  3746         1.160
 s        2        3978  3746         1.060
 tau      3        4112  3746         1.100
```

Ο έλεγχος σύγκλισης "Raftery and Lewis", όπως βλέπουμε στον *Πίνακα 3,* μας οδηγεί στο συμπέρασμα ότι η σύγκλιση είναι με μεγάλη πιθανότητα επιτυχής καθώς δεν φαίνεται να υπάρχει πρόβλημα αυτοσυσχετίσεων *(Dependence factors <= 1.18* **< 5** *).*

Πίνακας 3 "Πίνακας διαγνωστικού ελέγχου σύγκλισης 'Heidelberger and Welch'."

```
HEIDELBERGER AND WELCH STATIONARITY AND INTERVAL HALFWIDTH TESTS
================================================================

Iterations used = 2001:6000
Thinning interval = 1
Sample size per chain = 4000

Precision of halfwidth test = 0.1

$`C:/Users/part/Desktop/anova_all_models.out.txt`

         Stationarity start     p-value
         test         iteration
ab[2,2]  passed       1         0.463
ab[3,2]  passed       1         0.776
ab[4,2]  passed       1         0.699
ab[5,2]  passed       1         0.965
alpha[2] passed       1         0.376
beta[2]  passed       1         0.438
beta[3]  passed       1         0.835
beta[4]  passed       1         0.583
beta[5]  passed       1         0.999
m        passed       1         0.456
s        passed       1         0.104
tau      passed       1         0.120

         Halfwidth Mean     Halfwidth
         test
ab[2,2]  failed    9.84e+00 3.30e+00
ab[3,2]  passed    3.78e+01 2.99e+00
ab[4,2]  passed    7.20e+01 3.05e+00
ab[5,2]  passed    9.42e+01 2.99e+00
alpha[2] passed    2.29e+02 2.87e+00
beta[2]  passed    6.65e+01 3.04e+00
beta[3]  passed    2.00e+02 2.89e+00
beta[4]  passed    2.82e+02 2.91e+00
beta[5]  passed    5.66e+02 3.19e+00
m        passed    1.16e+03 3.43e+00
s        passed    1.83e+03 2.30e+00
tau      passed    3.00e-07 7.50e-10
```

Στον έλεγχο σύγκλισης "Heidelberger and Welch"ο οποίος φαίνεται στον *Πίνακα 3* που προηγείται, μας δείχνει ότι η υπόθεση της στασιμότητας της σειράς των προσομοιωμένων τιμών για τα διαφορετικά βήματα του αλγορίθμου πληρείται *(Stationarity test = **passed**)*. Αυτό σημαίνει ότι δεν υπάρχουν σημαντικές αυτοσυσχετίσεις και ο μέσος κυμαίνεται σε ένα σταθερό εύρος τιμών. Επομένως, η σύγκλιση είναι κατά πάσα πιθανότητα επιτυχής. Παρόλα αυτά, για την παράμετρο ab[2,2] παρατηρούμε ότι ο μέσος της posterior κατανομής δεν είναι τόσο ακριβής *(Halfwidth Mean test = failed)* και ίσως χρειάζεται να δοκιμάσουμε να τρέξουμε τον αλγόριθμό μας για μεγαλύτερο αριθμό επαναλήψεων.

Πίνακας 4 "Περιληπτικός πίνακας των περιγραφικών μέτρων των posterior κατανομών των παραμέτρων για το μοντέλο των κύριων επιδράσεων (main effects)."

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha[2]	245.1	89.27	2.036	69.11	245.5	419.9	4001	2000
beta[2]	71.45	98.34	1.784	-111.9	70.35	273.9	4001	2000
beta[3]	200.9	93.96	2.358	26.4	201.8	384.0	4001	2000
beta[4]	281.9	89.12	2.069	108.2	279.8	460.5	4001	2000
beta[5]	575.7	77.47	2.479	425.3	575.0	728.5	4001	2000
m	1168.0	76.1	2.639	1018.0	1167.0	1311.0	4001	2000
p.alpha[2]	0.9945	0.07396	0.001621	1.0	1.0	1.0	4001	2000
p.beta[2]	0.759	0.4277	0.00825	0.0	1.0	1.0	4001	2000
p.beta[3]	0.986	0.1175	0.003028	1.0	1.0	1.0	4001	2000
p.beta[4]	0.9995	0.02236	5.025E-4	1.0	1.0	1.0	4001	2000
p.beta[5]	1.0	0.0	2.236E-12	1.0	1.0	1.0	4001	2000
s	1829.0	60.14	1.715	1719.0	1828.0	1955.0	4001	2000

Ερμηνευτικά ή προβλεπτικά μοντέλα

Πίνακας 5 "Πίνακας διαγνωστικού ελέγχου σύγκλισης 'Raftery and Lewis'."

```
RAFTERY AND LEWIS CONVERGENCE DIAGNOSTIC
========================================

Iterations used = 5001:10000
Thinning interval = 1
Sample size per chain = 5000

$`C:\\Users\\part\\Desktop\\categ_int_1.out.txt`

Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

            Burn-in  Total Lower bound  Dependence
            (M)      (N)   (Nmin)       factor (I)
 alpha1[1]  3        4198  3746         1.120
 alpha1[2]  4        4713  3746         1.260
 alpha2[1]  3        4558  3746         1.220
 alpha2[2]  3        4410  3746         1.180
 alpha3[1]  2        3741  3746         0.999
 alpha3[2]  3        4129  3746         1.100
 alpha3[3]  2        3866  3746         1.030
 alpha3[4]  3        4293  3746         1.150
 alpha4[1]  2        3930  3746         1.050
 alpha4[2]  2        3803  3746         1.020
 alpha4[3]  2        3930  3746         1.050
 alpha4[4]  2        3620  3746         0.966
 alpha4[5]  4        4737  3746         1.260
 alpha5[1]  2        3829  3746         1.020
```

alpha5[2]	2	3803	3746	1.020
alpha5[3]	2	3803	3746	1.020
alpha5[4]	2	3866	3746	1.030
alpha5[5]	3	4484	3746	1.200
alpha6[1]	2	3680	3746	0.982
alpha6[2]	2	3741	3746	0.999
alpha6[3]	2	3741	3746	0.999
alpha6[4]	2	3680	3746	0.982
alpha6[5]	3	4558	3746	1.220
alpha7[1]	2	3995	3746	1.070
alpha7[2]	2	3866	3746	1.030
alpha7[3]	2	3829	3746	1.020
alpha7[4]	2	3620	3746	0.966
alpha7[5]	3	4558	3746	1.220
alpha8[1]	2	3561	3746	0.951
alpha8[2]	2	3803	3746	1.020
alpha8[3]	2	3741	3746	0.999
alpha8[4]	2	3620	3746	0.966
alpha8[5]	2	3741	3746	0.999
alpha8[6]	2	3620	3746	0.966
alpha8[7]	2	3930	3746	1.050
alpha8[8]	2	3741	3746	0.999
alpha8[9]	2	3741	3746	0.999
alpha8[10]	2	3930	3746	1.050
alpha9[1]	3	4198	3746	1.120
alpha9[2]	3	4198	3746	1.120
m	4	4873	3746	1.300
s	2	4010	3746	1.070

Από τον διαγγνωστικό έλεγχο σύγκλισης των Raftery and Lewis *(Πίνακας 5)*, συμπεραίνουμε ότι η σύγκλιση του αλγόριθμου MCM είναι με μεγάλη πιθανότητα επιτυχής (***Dependence Factor (I)*** *κοντά στη μονάδα*) για όλες τις παραμέτρους.

Πίνακας 6 "Deviance του μοντέλου των κύριων επιδράσεων."

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
deviance	20440.0	14.04	0.2328	20420.0	20440.0	20470.0	10001	5000

Πίνακας 7 "Deviance του μοντέλου των αλληλεπιδράσεων."

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
deviance	20440.0	13.98	0.2004	20420.0	20440.0	20470.0	5001	5000

Πίνακας 8 "Πίνακας διαγνωστικού ελέγχου σύγκλισης 'Raftery and Lewis' για το μοντέλο των αλληλεπιδράσεων με την μεταβλητή "shop"."

```
RAFTERY AND LEWIS CONVERGENCE DIAGNOSTIC
========================================

Iterations used = 5001:10000
Thinning interval = 1
Sample size per chain = 5000

$`C:/Users/part/Desktop/categ_int_1.out.txt`

Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

               Burn-in  Total Lower bound  Dependence
               (M)      (N)   (Nmin)       factor (I)
 alpha1[1]     2        3956  3746         1.060
 alpha1[2]     2        3803  3746         1.020
 alpha10[1,2]  2        3803  3746         1.020
 alpha10[1,3]  2        3620  3746         0.966
 alpha10[1,4]  2        3680  3746         0.982
 alpha10[2,1]  2        3620  3746         0.966
 alpha10[2,2]  2        3767  3746         1.010
 alpha10[2,3]  2        3741  3746         0.999
 alpha10[2,4]  2        3866  3746         1.030
 alpha10[3,1]  2        3995  3746         1.070
 alpha10[3,2]  2        3930  3746         1.050
 alpha10[3,3]  2        3803  3746         1.020
 alpha10[3,4]  2        3774  3746         1.010
 alpha10[4,1]  2        3866  3746         1.030
 alpha10[4,2]  2        3741  3746         0.999
 alpha10[4,3]  2        3680  3746         0.982
 alpha10[4,4]  2        3866  3746         1.030
 alpha10[5,1]  2        3803  3746         1.020
 alpha10[5,2]  2        3680  3746         0.982
 alpha10[5,3]  2        3706  3746         0.989
 alpha10[5,4]  2        3620  3746         0.966
 alpha10[6,1]  2        3741  3746         0.999
 alpha10[6,2]  2        3803  3746         1.020
 alpha10[6,3]  2        3620  3746         0.966
 alpha10[6,4]  2        3680  3746         0.982
 alpha10[7,1]  2        3653  3746         0.975
 alpha10[7,2]  2        3741  3746         0.999
 alpha10[7,3]  2        3706  3746         0.989
 alpha10[7,4]  2        3741  3746         0.999
 alpha10[8,1]  2        3892  3746         1.040
 alpha10[8,2]  2        3741  3746         0.999
 alpha10[8,3]  2        3680  3746         0.982
 alpha10[8,4]  2        3680  3746         0.982
 alpha10[9,1]  2        3741  3746         0.999
 alpha10[9,2]  2        3803  3746         1.020
 alpha10[9,3]  2        3680  3746         0.982
 alpha10[9,4]  2        3741  3746         0.999
 alpha10[10,1] 2        3646  3746         0.973
 alpha10[10,2] 2        3803  3746         1.020
 alpha10[10,3] 2        3620  3746         0.966
 alpha10[10,4] 2        3930  3746         1.050
 alpha11[1,2]  3        4267  3746         1.140
 alpha11[1,3]  2        3741  3746         0.999
```

alpha11[1,4]	2	3803	3746	1.020
alpha11[2,1]	2	3620	3746	0.966
alpha11[2,2]	2	3680	3746	0.982
alpha11[2,3]	2	3741	3746	0.999
alpha11[2,4]	2	3620	3746	0.966
alpha11[3,1]	2	3866	3746	1.030
alpha11[3,2]	2	3866	3746	1.030
alpha11[3,3]	2	3866	3746	1.030
alpha11[3,4]	2	3866	3746	1.030
alpha11[4,1]	2	3767	3746	1.010
alpha11[4,2]	2	3620	3746	0.966
alpha11[4,3]	2	3741	3746	0.999
alpha11[4,4]	2	3866	3746	1.030
alpha11[5,1]	2	3680	3746	0.982
alpha11[5,2]	2	3803	3746	1.020
alpha11[5,3]	2	3803	3746	1.020
alpha11[5,4]	2	3680	3746	0.982
alpha11[6,1]	2	3866	3746	1.030
alpha11[6,2]	2	3620	3746	0.966
alpha11[6,3]	2	3680	3746	0.982
alpha11[6,4]	2	3680	3746	0.982
alpha11[7,1]	2	3805	3746	1.020
alpha11[7,2]	2	3803	3746	1.020
alpha11[7,3]	2	3803	3746	1.020
alpha11[7,4]	2	3767	3746	1.010
alpha11[8,1]	2	3680	3746	0.982
alpha11[8,2]	2	3620	3746	0.966
alpha11[8,3]	2	3620	3746	0.966
alpha11[8,4]	2	3646	3746	0.973
alpha11[9,1]	2	3620	3746	0.966
alpha11[9,2]	2	3561	3746	0.951
alpha11[9,3]	2	3866	3746	1.030
alpha11[9,4]	2	3829	3746	1.020
alpha11[10,1]	2	3930	3746	1.050
alpha11[10,2]	2	3803	3746	1.020
alpha11[10,3]	2	3620	3746	0.966
alpha11[10,4]	2	3680	3746	0.982
alpha12[1,2]	2	3680	3746	0.982
alpha12[1,3]	2	3680	3746	0.982
alpha12[1,4]	2	3829	3746	1.020
alpha12[2,1]	2	3561	3746	0.951
alpha12[2,2]	2	3741	3746	0.999
alpha12[2,3]	2	3680	3746	0.982
alpha12[2,4]	2	3803	3746	1.020
alpha12[3,1]	2	3866	3746	1.030
alpha12[3,2]	2	3680	3746	0.982
alpha12[3,3]	2	3866	3746	1.030
alpha12[3,4]	2	3620	3746	0.966
alpha12[4,1]	2	3680	3746	0.982
alpha12[4,2]	2	3803	3746	1.020
alpha12[4,3]	2	3930	3746	1.050
alpha12[4,4]	2	3767	3746	1.010
alpha12[5,1]	2	3680	3746	0.982
alpha12[5,2]	2	3866	3746	1.030
alpha12[5,3]	2	3741	3746	0.999
alpha12[5,4]	2	3646	3746	0.973
alpha12[6,1]	2	3803	3746	1.020
alpha12[6,2]	2	3680	3746	0.982
alpha12[6,3]	2	3866	3746	1.030
alpha12[6,4]	2	3680	3746	0.982
alpha12[7,1]	2	3680	3746	0.982

alpha12[7,2]	2	3741	3746	0.999
alpha12[7,3]	2	3741	3746	0.999
alpha12[7,4]	2	3620	3746	0.966
alpha12[8,1]	2	3803	3746	1.020
alpha12[8,2]	2	3680	3746	0.982
alpha12[8,3]	2	3741	3746	0.999
alpha12[8,4]	2	3741	3746	0.999
alpha12[9,1]	2	3680	3746	0.982
alpha12[9,2]	2	3680	3746	0.982
alpha12[9,3]	2	3620	3746	0.966
alpha12[9,4]	2	3620	3746	0.966
alpha12[10,1]	2	3930	3746	1.050
alpha12[10,2]	2	3866	3746	1.030
alpha12[10,3]	2	3620	3746	0.966
alpha12[10,4]	2	3930	3746	1.050
alpha2[1]	3	4484	3746	1.200
alpha2[2]	2	3866	3746	1.030
alpha3[1]	2	3995	3746	1.070
alpha3[2]	3	4155	3746	1.110
alpha3[3]	2	3995	3746	1.070
alpha3[4]	2	3930	3746	1.050
alpha4[1]	2	3866	3746	1.030
alpha4[2]	2	3741	3746	0.999
alpha4[3]	2	3741	3746	0.999
alpha4[4]	2	3741	3746	0.999
alpha4[5]	4	4660	3746	1.240
alpha5[1]	3	4267	3746	1.140
alpha5[2]	2	3741	3746	0.999
alpha5[3]	2	3930	3746	1.050
alpha5[4]	2	3561	3746	0.951
alpha5[5]	4	4713	3746	1.260
alpha6[1]	3	4155	3746	1.110
alpha6[2]	2	3995	3746	1.070
alpha6[3]	2	3681	3746	0.983
alpha6[4]	2	3680	3746	0.982
alpha6[5]	3	4410	3746	1.180
alpha7[1]	2	3995	3746	1.070
alpha7[2]	2	3866	3746	1.030
alpha7[3]	2	3892	3746	1.040
alpha7[4]	2	3620	3746	0.966
alpha7[5]	3	4198	3746	1.120
alpha8[1]	2	3620	3746	0.966
alpha8[2]	2	3866	3746	1.030
alpha8[3]	2	3803	3746	1.020
alpha8[4]	2	3803	3746	1.020
alpha8[5]	2	3594	3746	0.959
alpha8[6]	2	3995	3746	1.070
alpha8[7]	2	3680	3746	0.982
alpha8[8]	2	4020	3746	1.070
alpha8[9]	2	3866	3746	1.030
alpha8[10]	2	3995	3746	1.070
alpha9[1,2]	2	3620	3746	0.966
alpha9[1,3]	2	3561	3746	0.951
alpha9[1,4]	2	3803	3746	1.020
alpha9[2,1]	2	3866	3746	1.030
alpha9[2,2]	2	3803	3746	1.020
alpha9[2,3]	2	3561	3746	0.951
alpha9[2,4]	2	3680	3746	0.982
alpha9[3,1]	2	3803	3746	1.020
alpha9[3,2]	2	3620	3746	0.966
alpha9[3,3]	2	3680	3746	0.982

alpha9[3,4]	2	3767	3746	1.010
alpha9[4,1]	2	3866	3746	1.030
alpha9[4,2]	2	3741	3746	0.999
alpha9[4,3]	2	3620	3746	0.966
alpha9[4,4]	2	3680	3746	0.982
alpha9[5,1]	2	3620	3746	0.966
alpha9[5,2]	2	3588	3746	0.958
alpha9[5,3]	2	3680	3746	0.982
alpha9[5,4]	2	3588	3746	0.958
alpha9[6,1]	2	3620	3746	0.966
alpha9[6,2]	2	3706	3746	0.989
alpha9[6,3]	2	3930	3746	1.050
alpha9[6,4]	2	3866	3746	1.030
alpha9[7,1]	2	3680	3746	0.982
alpha9[7,2]	2	3866	3746	1.030
alpha9[7,3]	2	3620	3746	0.966
alpha9[7,4]	2	3706	3746	0.989
alpha9[8,1]	2	3930	3746	1.050
alpha9[8,2]	2	3680	3746	0.982
alpha9[8,3]	2	3706	3746	0.989
alpha9[8,4]	2	3680	3746	0.982
alpha9[9,1]	2	3594	3746	0.959
alpha9[9,2]	2	3803	3746	1.020
alpha9[9,3]	2	3866	3746	1.030
alpha9[9,4]	2	3620	3746	0.966
alpha9[10,1]	2	3680	3746	0.982
alpha9[10,2]	2	3561	3746	0.951
alpha9[10,3]	2	3866	3746	1.030
alpha9[10,4]	2	3803	3746	1.020
m	4	5211	3746	1.390
s	3	4325	3746	1.150

Από τον έλεγχο σύγκλισης “Raftery and Lewis” (*Πίνακα 8),* συμπεραίνουμε ότι η σύγκλιση είναι με μεγάλη πιθανότητα επιτυχής *(Dependence factors κοντά στο 1)*.

Πίνακας 9 “Deviance του μοντέλου των αλληλεπιδράσεων με την μεταβλητή “shop”.”

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
deviance	20430.0	15.19	0.232	20410.0	20430.0	20470.0	5001	5000

DIC = 0.5 $(15.19)^2$ +20430 = 20546

Πίνακας 10 “Deviance του μοντέλου “random effects”.”

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
deviance	20440.0	14.95	0.2684	20410.0	20440.0	20470.0	5001	5000

DIC = 0.5 $(14.95)^2$ + 20440 = 20552

Πίνακας 11 "Deviance του μοντέλου των κατηγορικών μεταβλητών κυρίων επιδράσεων, χωρίς τη μεταβλητή "shop"."

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
deviance	18140.0	13.41	0.1808	18120.0	18140.0	18170.0	10001	5000

DIC = $0.5\,(13.41)^2 + 18140 = 18230$

Πίνακας 12 "Deviance του μοντέλου των ποσοτικών και κατηγορικών συμμεταβητών (ANCOVA)."

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
deviance	18240.0	19.17	0.7864	18210.0	18240.0	18290.0	5001	1000

DIC = $0.5\,(19.17)^2 + 18240 = 18424$

Πίνακας 13 "Πίνακας περιγραφικών μέτρων τελικού μοντέλου"

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha1[1]	-304.1	61.3	0.8679	-421.8	-304.9	-183.7	5001	6000
alpha1[2]	304.1	61.3	0.8679	184.0	305.0	422.0	5001	6000
alpha2[1]	143.5	63.15	0.9357	20.51	143.8	266.8	5001	6000
alpha2[2]	-143.5	63.15	0.9357	-266.7	-143.8	-20.47	5001	6000
alpha3[1]	68.17	106.1	1.285	-136.2	68.75	273.5	5001	6000
alpha3[2]	262.0	70.28	1.196	121.4	261.2	400.3	5001	6000
alpha3[3]	-123.4	86.65	1.522	-289.5	-123.2	48.17	5001	6000
alpha3[4]	-206.8	89.74	1.532	-383.0	-207.7	-31.5	5001	6000
alpha4[1]	-471.1	174.6	2.468	-809.3	-470.0	-133.5	5001	6000
alpha4[2]	-33.38	94.95	1.227	-223.0	-33.96	153.0	5001	6000
alpha4[3]	5.776	92.4	1.128	-175.0	6.383	186.2	5001	6000
alpha4[4]	99.58	88.86	1.181	-74.82	100.5	271.4	5001	6000
alpha4[5]	399.1	81.08	1.496	239.0	399.4	559.0	5001	6000
alpha5[1]	-399.7	178.5	2.415	-745.5	-400.5	-57.63	5001	6000
alpha5[2]	-39.38	94.82	1.114	-225.2	-39.66	142.7	5001	6000
alpha5[3]	44.76	88.46	1.038	-123.4	44.4	222.4	5001	6000
alpha5[4]	128.3	87.28	1.248	-42.8	127.3	299.9	5001	6000
alpha5[5]	265.9	81.24	1.39	106.9	266.6	426.8	5001	6000
alpha6[1]	-478.6	171.8	2.29	-818.5	-476.0	-147.6	5001	6000
alpha6[2]	15.02	95.02	1.177	-170.4	14.4	198.2	5001	6000
alpha6[3]	110.9	87.86	1.186	-64.41	109.6	282.8	5001	6000
alpha6[4]	145.9	87.4	1.262	-22.19	144.4	320.0	5001	6000
alpha6[5]	206.9	80.55	1.464	52.06	207.2	365.2	5001	6000
alpha7[1]	-377.3	173.3	2.214	-714.9	-376.6	-37.29	5001	6000
alpha7[2]	-22.67	95.98	1.352	-209.5	-23.49	165.3	5001	6000
alpha7[3]	76.49	89.43	1.184	-98.79	77.35	251.8	5001	6000
alpha7[4]	142.6	86.76	1.174	-28.87	141.9	318.1	5001	6000
alpha7[5]	180.9	80.43	1.23	22.08	180.9	335.7	5001	6000
m	802.4	81.64	1.729	640.5	802.8	960.5	5001	6000

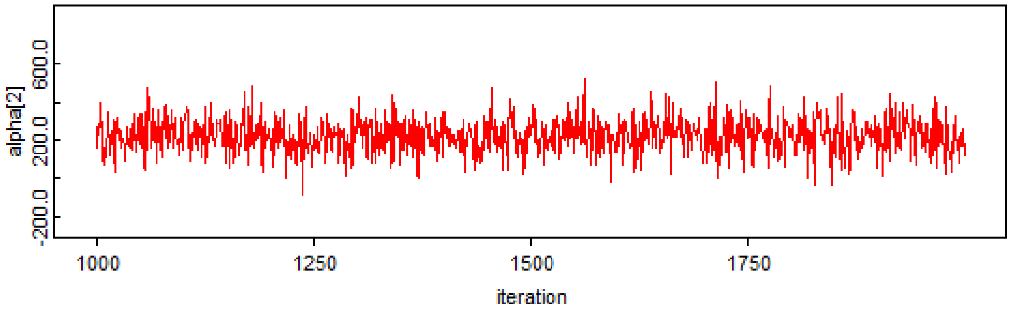
Πίνακας 14 "Πίνακας των p0 του τελικού μοντέλου"

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
p.alpha1[2]	1.0	0.0	3.162E-12	1.0	1.0	1.0	10001	1000
p.alpha2[2]	0.016	0.1255	0.003511	0.0	0.0	0.0	10001	1000
p.alpha3[1]	0.722	0.448	0.008246	0.0	1.0	1.0	10001	1000
p.alpha3[2]	1.0	0.0	3.162E-12	1.0	1.0	1.0	10001	1000
p.alpha3[3]	0.082	0.2744	0.009338	0.0	0.0	1.0	10001	1000
p.alpha3[4]	0.01	0.0995	0.003032	0.0	0.0	0.0	10001	1000
p.alpha4[1]	0.005	0.07053	0.002082	0.0	0.0	0.0	10001	1000
p.alpha4[2]	0.346	0.4757	0.01197	0.0	0.0	1.0	10001	1000
p.alpha4[3]	0.534	0.4988	0.01716	0.0	1.0	1.0	10001	1000
p.alpha4[4]	0.881	0.3238	0.009366	0.0	1.0	1.0	10001	1000
p.alpha4[5]	1.0	0.0	3.162E-12	1.0	1.0	1.0	10001	1000
p.alpha5[1]	0.01	0.0995	0.002642	0.0	0.0	0.0	10001	1000
p.alpha5[2]	0.342	0.4744	0.01489	0.0	0.0	1.0	10001	1000
p.alpha5[3]	0.696	0.46	0.01216	0.0	1.0	1.0	10001	1000
p.alpha5[4]	0.947	0.224	0.006504	0.0	1.0	1.0	10001	1000
p.alpha5[5]	1.0	0.0	3.162E-12	1.0	1.0	1.0	10001	1000
p.alpha6[1]	0.003	0.05469	0.001665	0.0	0.0	0.0	10001	1000
p.alpha6[2]	0.555	0.497	0.01993	0.0	1.0	1.0	10001	1000
p.alpha6[3]	0.893	0.3091	0.008468	0.0	1.0	1.0	10001	1000
p.alpha6[4]	0.941	0.2356	0.00672	0.0	1.0	1.0	10001	1000
p.alpha6[5]	0.997	0.05469	0.001665	1.0	1.0	1.0	10001	1000
p.alpha7[1]	0.014	0.1175	0.003194	0.0	0.0	0.0	10001	1000
p.alpha7[2]	0.422	0.4939	0.01729	0.0	0.0	1.0	10001	1000
p.alpha7[3]	0.788	0.4087	0.01131	0.0	1.0	1.0	10001	1000
p.alpha7[4]	0.943	0.2318	0.006616	0.0	1.0	1.0	10001	1000
p.alpha7[5]	0.989	0.1043	0.003084	1.0	1.0	1.0	10001	1000
s	1713.0	52.57	0.849	1615.0	1711.0	1821.0	5001	6000

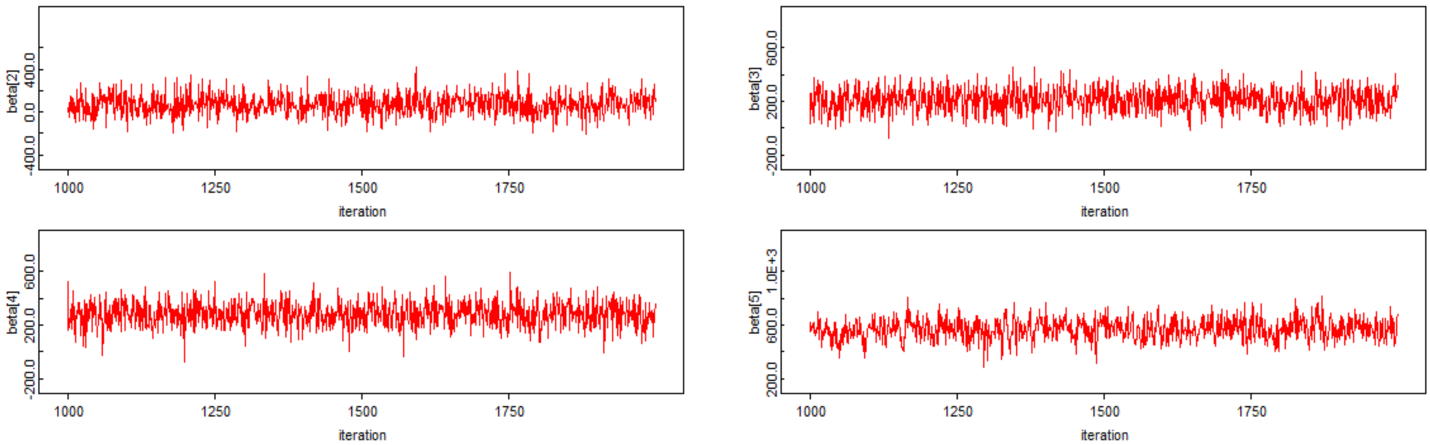
Διαγράμματα

Ανάλυση Διακύμανσης (ANOVA)

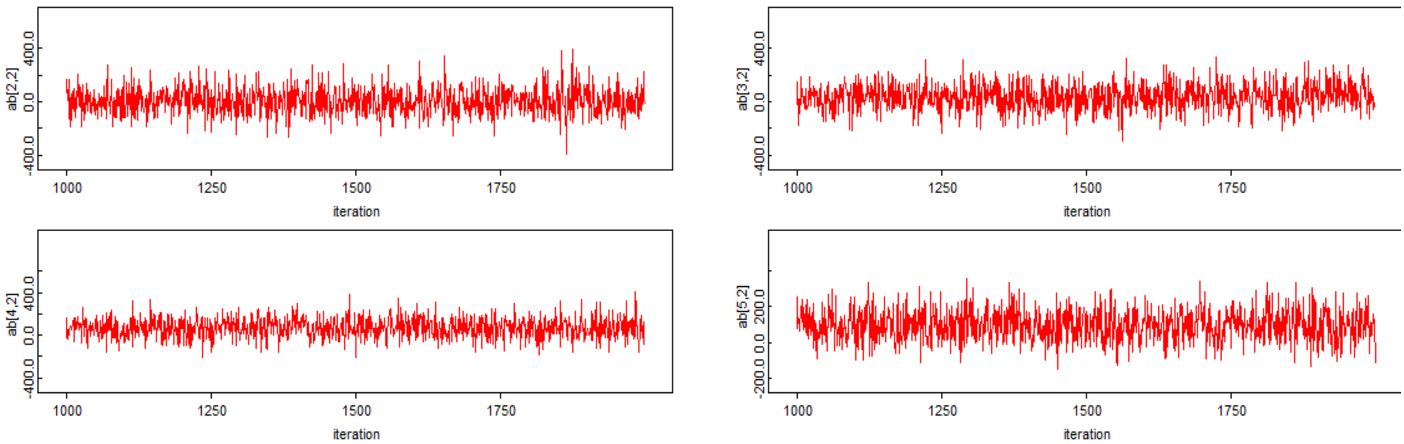
Διάγραμμα 1.α "Trace plot παραμέτρου alpha[2] για το μοντέλο των αλληλεπιδράσεων"



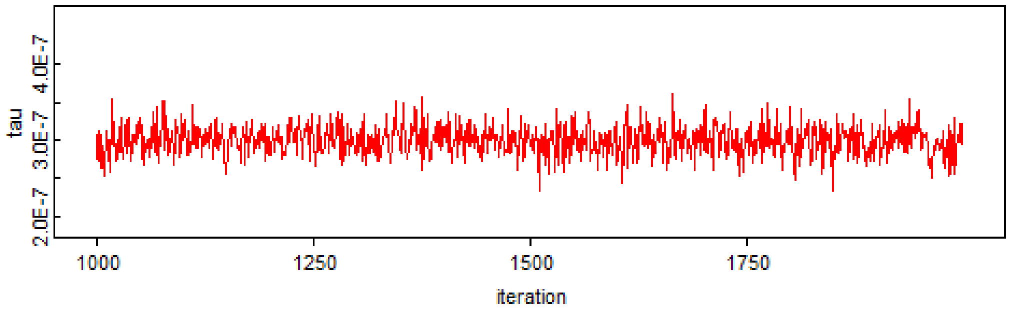
Διάγραμμα 1.β "Trace plots παραμέτρων beta[2], beta[3], beta[4], beta[5] για το μοντέλο των αλληλεπιδράσεων"



Διάγραμμα 1.γ "Trace plots παραμέτρων ab[2,2], ab[3,2], ab[4,2], ab[5,2] για το μοντέλο των αλληλεπιδράσεων"

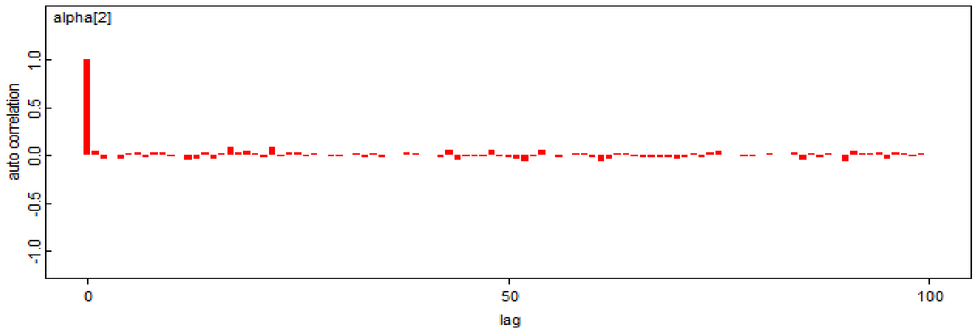


Διάγραμμα 1.γ "Trace plot παραμέτρου tau για το μοντέλο των αλληλεπιδράσεων"

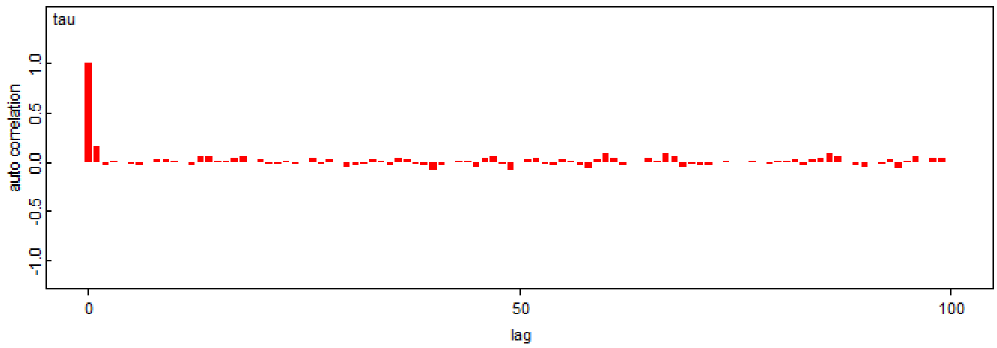


Από τα διαγράμματα *1.α – 1.γ* , παρατηρούμε ότι η διακύμανση των posterior κατανομών των παραμέτρων μας είναι σταθερή χωρίς κάποιο "peak" και ο αλγόριθμος MCMC συγκλίνει ομαλά.

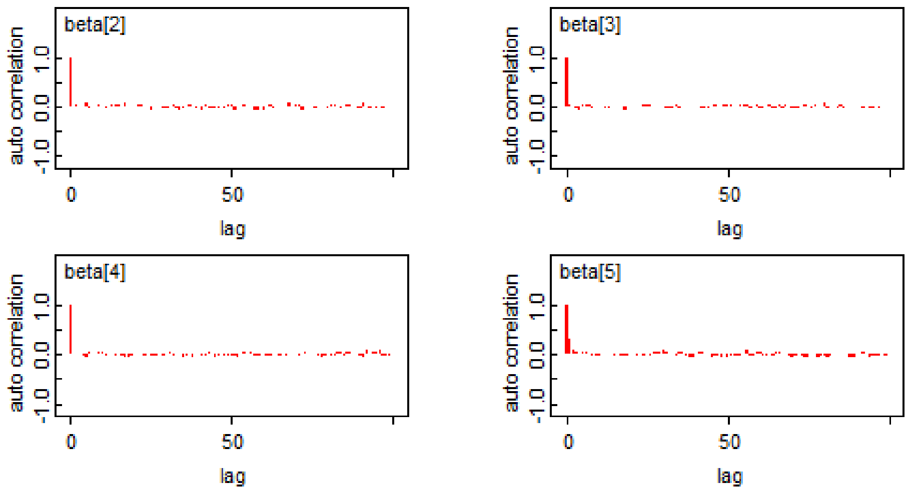
Διάγραμμα 2.α "Autocorrelations plot παραμέτρου alpha[2] για το μοντέλο των αλληλεπιδράσεων"



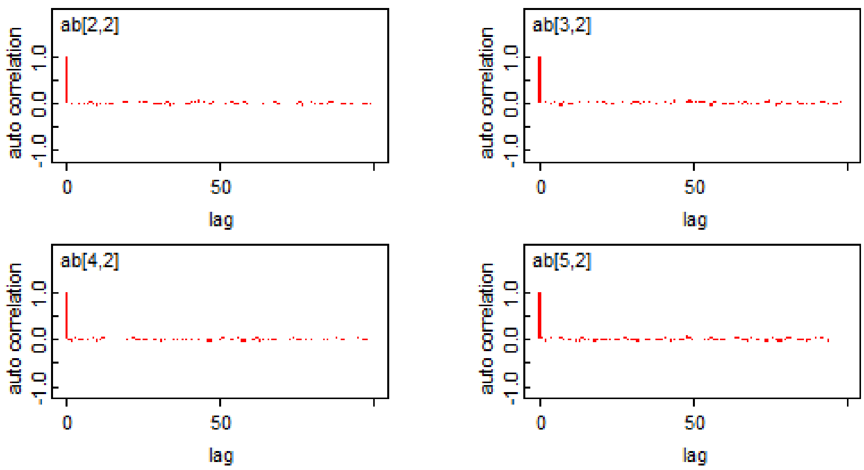
Διάγραμμα 2.β "Autocorrelations plot παραμέτρου tau για το μοντέλο των αλληλεπιδράσεων"



Διάγραμμα 2.γ “Autocorrelations plot παραμέτρων beta[2], beta[3], beta[4], beta[5] για το μοντέλο των αλληλεπιδράσεων”

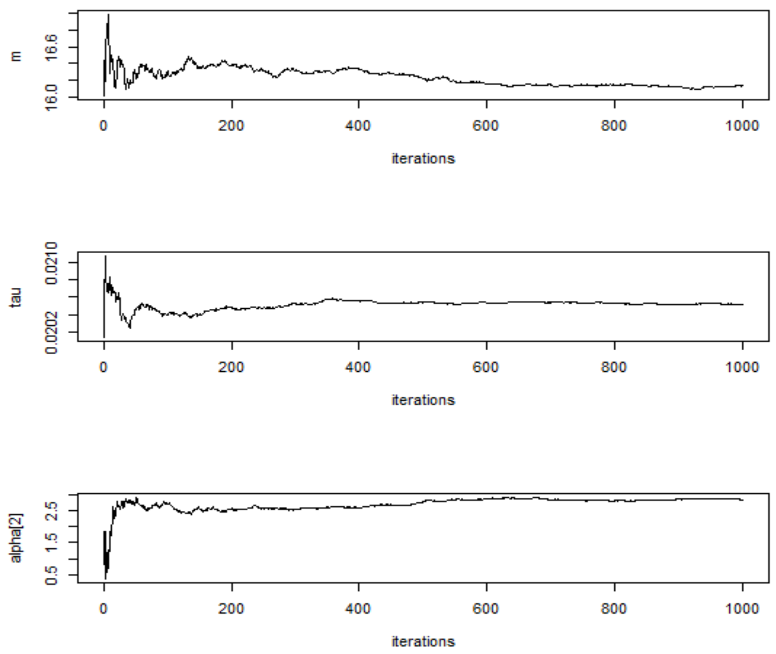


Διάγραμμα 2.γ “Autocorrelations plot παραμέτρων ab[2,2], ab[3,2], ab[4,2], ab[5,2] για το μοντέλο των αλληλεπιδράσεων”

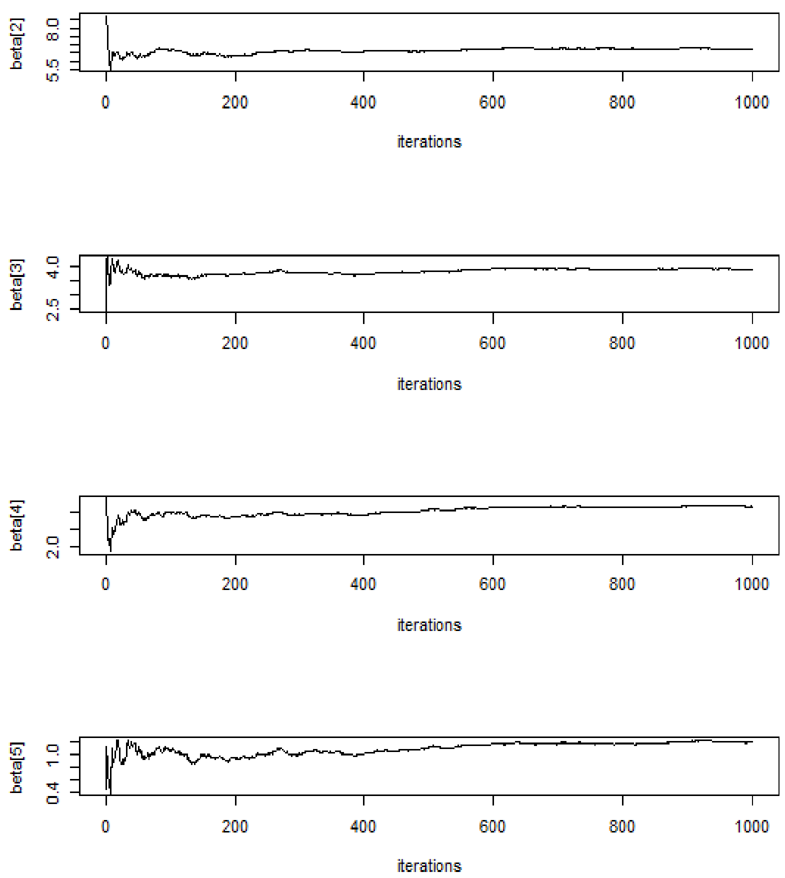


Τα autocorrelations plots *(Διαγράμματα 2.α – 2.γ.)* μας επιβεβαιώνουν ότι η σύγκλιση γίνεται κανονικά χωρίς ύπαρξη αυτοσυσχετίσεων, καθώς όπως παρατηρούμε οι συντελεστές αυτοσυσχέτισης όλων των παραμέτρων φθίνουν γρήγορα στο μηδέν.

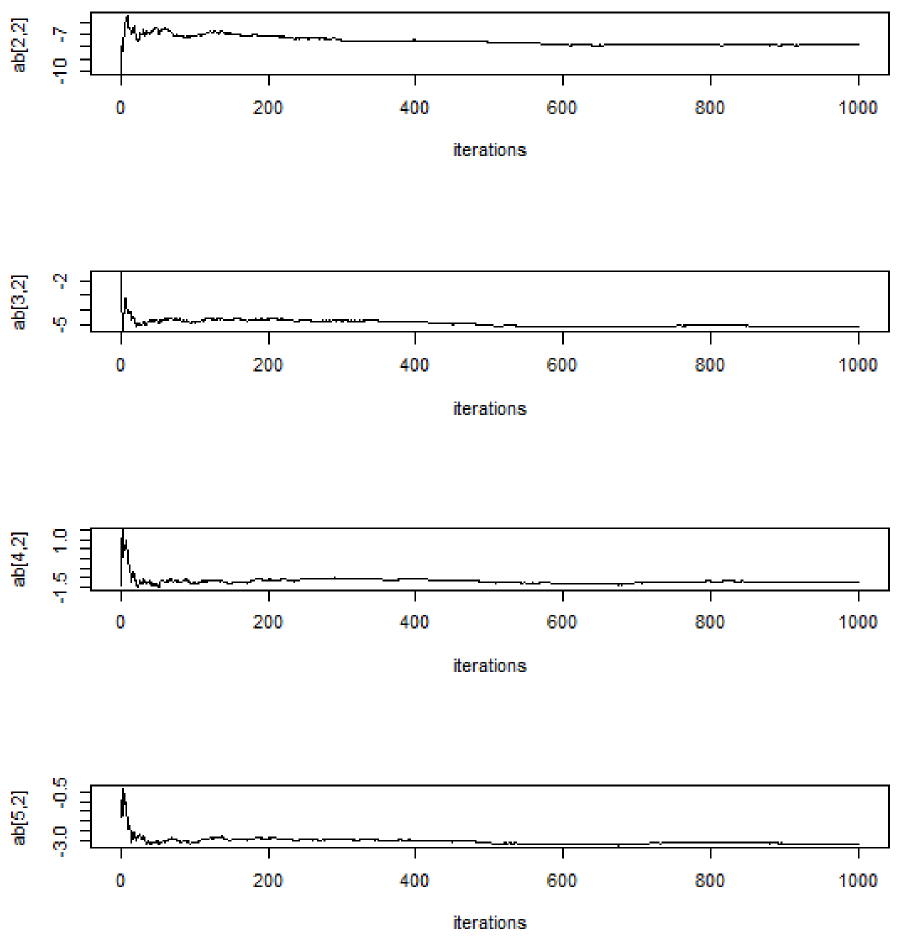
Διάγραμμα 3.α “Egotic mean plots των παραμέτρου alpha[2], m, tau για το μοντέλο των αλληλεπιδράσεων”



Διάγραμμα 3.β “Egotic mean plots των παραμέτρου beta[2], beta[3], beta[4], beta[5] για το μοντέλο των αλληλεπιδράσεων”

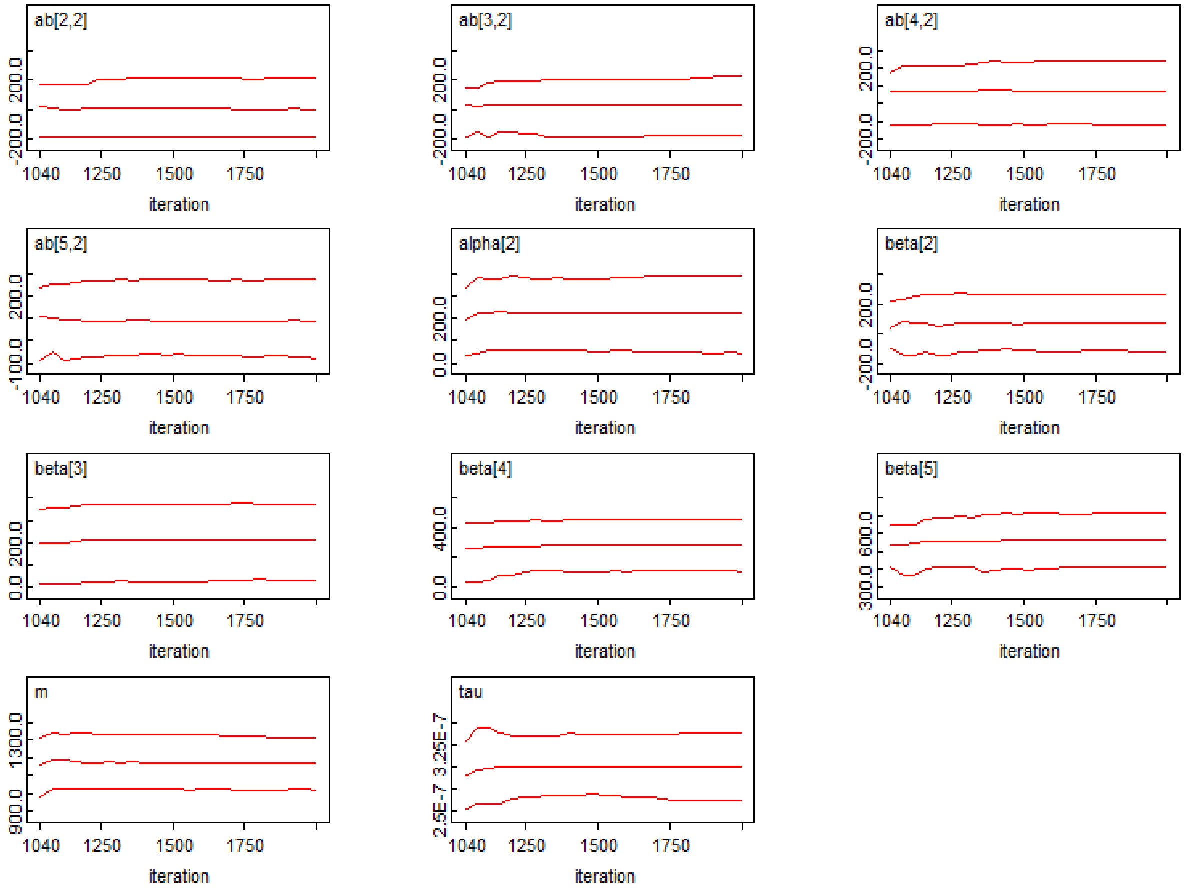


Διάγραμμα 3.γ “Egotic mean plots των παραμέτρου ab[2,2], ab[3,2], ab[4,2], ab[5,2] για το μοντέλο των αλληλεπιδράσεων”



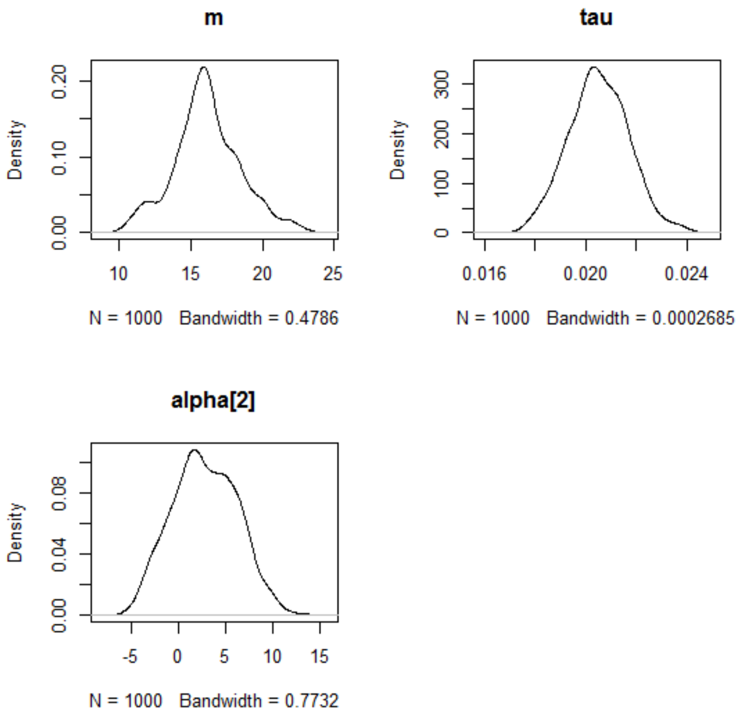
Παρατηρούμε σταθεροποίηση της τιμής του μέσου των posterior κατανομών όλων των παραμέτρων, ως προς τον συνολικό αριθμό επαναλήψεων κάθε βήματος σχετικά γρήγορα *(Διαγράμματα 3.α-3.γ)*. Συνεπώς, δεν φαίνεται να έχουμε πρόβλημα σύγκλισης.

Διάγραμμα 4 “Quantiles plots των παραμέτρων του μοντέλου των αλληλεπιδράσεων”

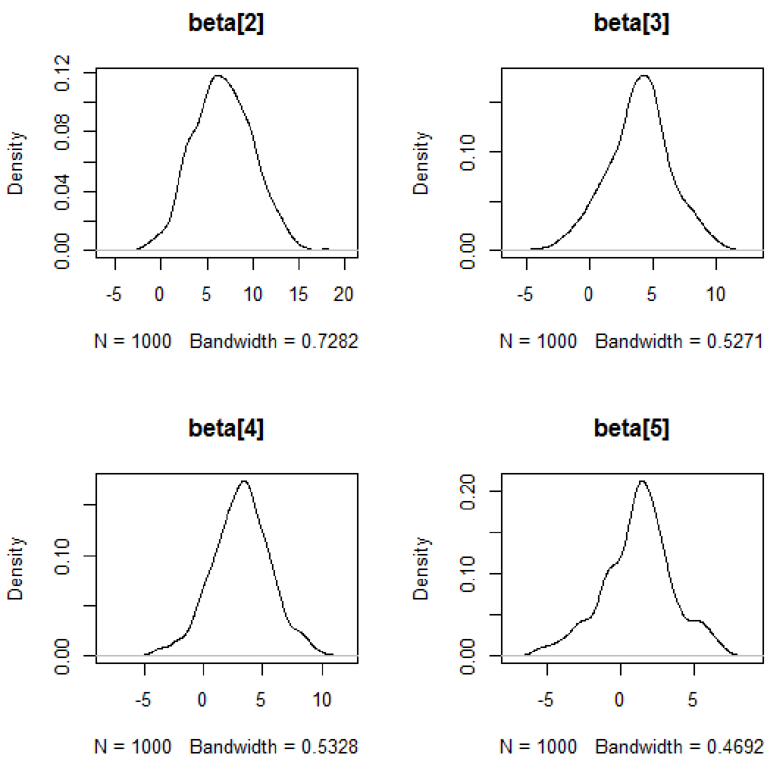


Στο διάγραμμα 4 , παρατηρούμε ότι όλες οι τιμές του κινούμενου μέσου της posterior κατανομής των παραμέτρων και τα διαστήματα εμπιστοσύνης είναι σταθερά, επομένως έχουμε ισχυρές ενδείξεις ότι η σύγκλιση γίνεται κανονικά. Επίσης, τα 95% διαστήματα εμπιστοσύνης είναι αρκετά κλειστά υποδηλώνοντας ένα χαμηλό Monte Carlo error και μεγάλη ακρίβεια στις εκτιμήσεις

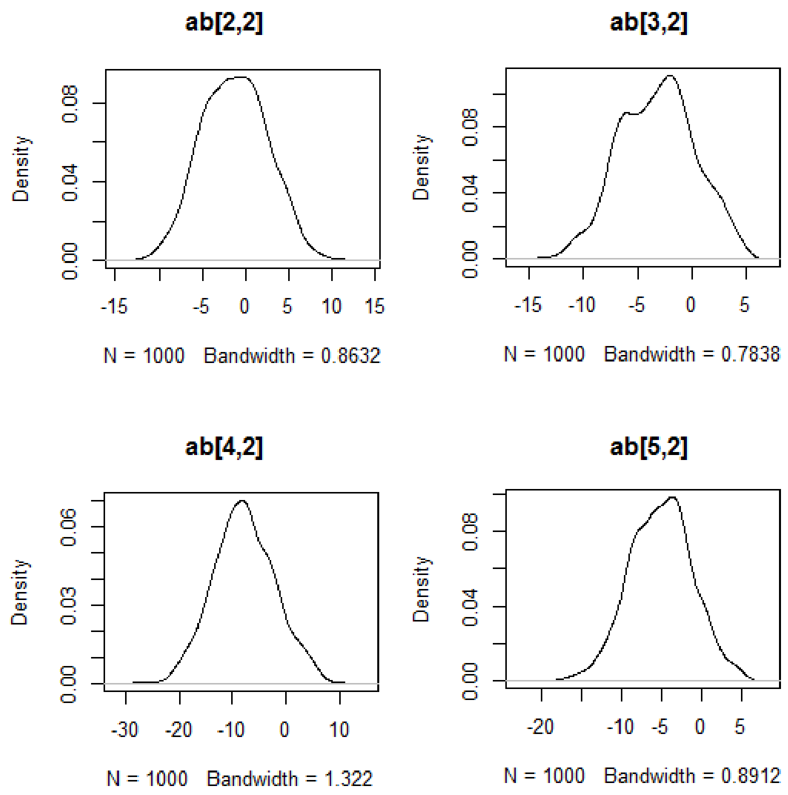
Διάγραμμα 5.α "density plots των παραμέτρου alpha[2], m, tau για το μοντέλο των αλληλεπιδράσεων"



Διάγραμμα 5.β "density plots των παραμέτρου beta[2], beta[3], beta[4], beta[5] για το μοντέλο των αλληλεπιδράσεων"

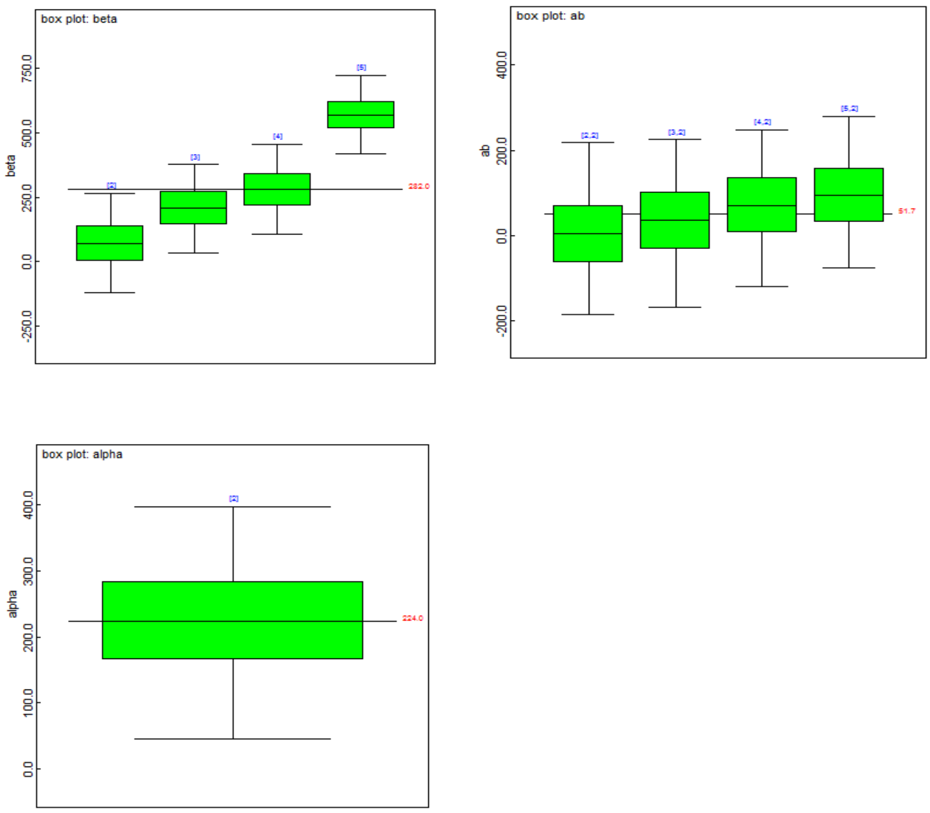


Διάγραμμα 5.γ “density plots των παραμέτρου ab[2,2], ab[3,2], ab[4,2], ab[5,2] για το μοντέλο των αλληλεπιδράσεων”



Από τα διαγράμματα πυκνότητας πιθανότητας των posterior κατανομών των παραμέτρων (*Διαγράμματα 4.α-4.γ*) , θα λέγαμε ότι οι κατανομές φαίνονται συμμετρικές χωρίς ιδιαίτερες αποκλίσεις από την κανονική κατανομή και μονοκόρυφες, επομένως μπορούμε να χρησιμοποιήσουμε τον κλασσικό τύπο του κριτηρίου DIC χωρίς πρόβλημα. Το 0 είναι στα άκρα της κατανομής των παραμέτρων επίδρασης, με εξαίρεση την παραμέτρους ab[2,2] η οποία αναμένουμε να μην έχει στατιστικά σημαντική επίδραση στη μεταβλητή ενδιαφέροντος.

Διάγραμμα 6 "Box-plots των παραμέτρων επίδρασης για το μοντέλο των αλληλεπιδράσεων"



Από το *Διάγραμμα 6*, συμπεραίνουμε ότι η παράμετρος alpha[2] είναι με μεγάλη πιθανότητα στατιστικά σημαντική καθώς δείχνει μεγάλη απόκλιση από το 0 το οποίο βρίσκεται εκτός του πεδίου τιμών της και το αναμενόμενο ποσό χρημάτων που δαπανάται από άντρες καταναλωτές διαφέρει σημαντικά από το αναμενόμενο ποσό που δαπανάται από γυναίκες.

Επίσης, στατιστικά σημαντικές φαίνονται και όλες οι παράμετροι επίδρασης του εκτιμώμενου επιπέδου τιμών (beta[2,2], beta[3,2], beta[4,2], beta[5,2]). Ειδικότερα, η παράμετρος που αφορά την επίδραση του επιπέδου 5-"πολύ χαμήλές τιμές" φαίνεται ιδιαίτερα ισχυρή και λιγότερο ισχυρή η επίδραση του επιπέδου 2 -"υψηλές τιμές", όπου το 0 βρίσκεται στα άκρα του διαστήματος εμπιστοσύνης.

Επιπλέον, συγκρίνοντας τα box-plots των παραμέτρων επίδρασης της τιμής μπορούμε να πούμε ότι το μέσο χρηματικό ποσό που δαπανάται διαφέρει σημαντικά ανάμεσα σε όλα τα διαφορετικά επίπεδα τιμών. Μικρή επικάλυψη παρατηρούμε μόνο στα διαστήματα εμπιστοσύνης των παραμέτρων των επιπέδων 2 και 3, και μεγαλύτερη μεταξύ των επιπέδων 3 και 4, επομένως δεν φαίνεται να υπάρχουν ιδιαίτερες διαφορές μεταξύ των δύο. Γενικά, παρατηρούμε ότι όσο αυξάνεται το επίπεδο τιμών, δηλαδή οι τιμές εκτιμώνται χαμηλότερες, τόσο το αναμενόμενο ποσό χρημάτων που δαπανάται αυξάνεται.

Τέλος, η αλληλεπίδραση του επιπέδου των τιμών με τη μεταβλητή του φύλου (ab[,]) δε φαίνεται στατιστικά σημαντική καθώς το 0 είναι κεντρικό σημείο της κατανομής και σχεδόν ταυτίζεται με το μέσο στην παράμετρο ab[2,2], ενώ το 0 βρίσκεται σε κεντρικό σημείο και στις υπόλοιπες παραμέτρους (ab[3,2],ab[4,2],ab[5,2]). Επομένως, ο όρος της αλληλεπίδρασης ενδεχομένως να μη χρειάζεται να χρησιμοποιηθεί στο μοντέλο μας.