Assignment 3

Andy Yuan

2025-04-02

Question 1

Data is generated from the exponential distribution with density $f(y) = \lambda exp(-\lambda y)$ where $\lambda, y > 0$

a)

To convert this function to the general form of the exponential family, we take the coefficient λ up inside the exponent.

$$f(y) = exp(log\lambda - \lambda y)$$

Now, we can identify the various parameters:

- $\theta = -\lambda$
- $b(\theta) = -log(\lambda)$
- $\phi = 1$
- $a(\phi) = 1$
- $c(y, \phi) = 0$

b)

The mean of the exponential distribution is $1/\lambda$

The canonical link is $-\lambda$, which equals $-1/\mu$ since the canonical link should be expressed in terms of μ

In a similar vein, the variance function $V(\mu) = b''(\theta)/w$ is:

$$V(\mu) = b''(\theta) = -1/\lambda^2 = \mu^2$$

- $w = 1$ since $a(\phi) = \phi/w = 1$

c)

Since the canonical link depends on μ, where μ is the denominator, and is negative, this can lead to convergence issues in GLM fits.

d)

Since there is a dispersion parameter that is known ($\phi = 1$), we can use the chi-squared tests since the deviance and pearson chi-squared test statistics are asymptotically chi-squared distributed.

e)

Assuming that the observations are independent and identically distributed according to the density function given, the likelihood and the log-likelihood is then shown below. Where:

- i is the index of the points
- n is the total number of observations

$$L = \prod_{i=1}^{n} \hat{\mu}_i exp(-\hat{\mu}_i y_i)$$

$$logL = \sum_{i=1}^{n} (log\hat{\mu}_i - \hat{\mu}_i y_i)$$

$$\text{In a saturated model, } y_i = 1/\hat{\mu}_i \text{, since } dlogL/d\mu_i = 1/\mu_i - y_i.$$

$$\text{Then when we set to 0, } y_i = 1/\hat{\mu}_i$$

$$\text{In a fitted model, we assume } \hat{\mu} = 1/\bar{y}$$

$$\begin{aligned} logL_{Saturated} &= \sum_{i=1}^{n} (-logy_i - 1) \\ logL_{model} &= \sum_{i=1}^{n} (log(1/\bar{y}) - y_i/\bar{y}) \\ &= -nlog\bar{y} - n \end{aligned}$$

Thus the deviance is:

$$\begin{aligned} D &= 2(\sum_{i=1}^{n} (-logy_i - 1) - (-nlog\bar{y} - n)) \\ &= 2(\sum_{i=1}^{n} (-logy_i) - n + nlog\bar{y} + n) \\ &= 2(\sum_{i=1}^{n} (-logy_i) + nlog\bar{y}) \\ &= 2\sum_{i=1}^{n} (log(\bar{y}/y_i)) \end{aligned}$$

Question 2

Consider the Galápagos data and model analyzed in this chapter. The purpose of this question is to reproduce the details of the GLM fitting of this data.

a)

```
suppressMessages(library(faraway))
data(gala)
str(gala)
```

```
## 'data.frame':    30 obs. of  7 variables:
##  $ Species  : num  58 31 3 25 2 18 24 10 8 2 ...
##  $ Endemics : num  23 21 3 9 1 11 0 7 4 2 ...
##  $ Area     : num  25.09 1.24 0.21 0.1 0.05 ...
##  $ Elevation: num  346 109 114 46 77 119 93 168 71 112 ...
##  $ Nearest  : num  0.6 0.6 2.8 1.9 1.9 8 6 34.1 0.4 2.6 ...
##  $ Scruz    : num  0.6 26.3 58.7 47.4 1.9 ...
##  $ Adjacent : num  1.84 572.33 0.78 0.18 903.82 ...
```

```
modpois <- glm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
               data=gala, family=poisson)
summary(modpois)
```

```
##
## Call:
## glm(formula = Species ~ Area + Elevation + Nearest + Scruz +
##     Adjacent, family = poisson, data = gala)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
## Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
## Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
## Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
## Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
## Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  716.85  on 24  degrees of freedom
## AIC: 889.68
##
## Number of Fisher Scoring iterations: 5
```

The deviance is 716.85, the coefficients are:

- intercept: 3.155e+00
- area: -5.799e-04
- elevation: 3.541e-03
- nearest: 8.826e-03
- scruz: -5.709e-03
- adjacent: -6.630e-04

b)

For the poisson GLM, $f(y|\mu,\phi) = exp(ylog\mu - \mu - logy!)$

Where:

- $\theta = log\mu$: the canonical parameter
- $\phi = 1$: the dispersion parameter
- $a(\phi) = 1$
- $b(\theta) = exp(\theta)$
- $c(y,\phi) = -logy!$
- $\tilde{\theta}_i = logy_i$
- $\hat{\theta}_i = log\hat{\mu}_i$
- $b(\tilde{\theta}_i) = y_i$
- $b(\hat{\theta}_i) = \hat{\mu}_i$

Showing eta

η is how the mean μ relates to the covariates and their coefficients. I will show it below where L represents the likelihood, μ represents the mean and variance of the distribution, and y represents the response variable:

$$\begin{aligned} L &= \prod_{i=1}^{n} \frac{e^{-\mu}\mu^{y_i}}{y_i!} \\ logL &= \sum_{i=1}^{n} (y_i log\mu - \mu - logy_i!) \\ \eta &= log\mu, \text{since it is the canonical link} \end{aligned}$$

The canonical link has g such that $\eta = g(\mu) = \theta$, so the canonical link of the poisson distribution is $log\mu$

The variance of the mean

$$Var(Y) = b''(\theta)a(\phi) = V(\mu)\phi$$

Knowing this definition, we can use previous definitions to answer what $V(\mu)$ is:

$$b(\theta) = b'(\theta) = b''(\theta) = \mu, \phi = 1, a(\phi) = 1$$

Therefore, the variance of the mean $V(\mu) = \mu$.

The derivative of eta wrt mu

Since our η already has μ, we don't need to do any transformations.

$$\begin{aligned} \eta &= log\mu \\ \partial\eta/\partial\mu &= \partial/\partial\mu(log\mu) \\ &= 1/\mu \end{aligned}$$

Weights:

the weights are: $\frac{1}{V(\mu_i^{(c-1)})}\left[\frac{\partial\eta_i}{\partial\mu_i}\right]^{-2}\Bigg|_{\mu_i^{(c-1)}}$, which in our current framework, holds as: $\frac{1}{\mu\times(1/\mu)^2} = \mu$

Adjusted dependent variable

The adjusted dependent variable, $z^{(c-1)} = \eta^{(c-1)} + M^{(c-1)}(y - \mu^{(c-1)})$, where $M^{(c-1)} = \frac{\partial\eta_i}{\partial\mu_i}\Big|_{\mu_i^{(c-1)}}$, is $log\hat{\mu} + \frac{1}{\mu}(y - \hat{\mu})$

c)

The adjusted dependent variable is:

$$z^c = \hat{\eta}^c + M^c(y - \hat{\mu}^c)$$

The weight matrix is:

$$W^c = \frac{1}{V(\mu^c)}\left(\frac{\partial\eta}{\partial\mu}\right)^{-2}\Bigg|_{\mu^c}$$

The next β is:

$$\beta^{c+1} = [X^TW^cX]^{-1}X^TW^cz^c$$

So for our algorithm:

- $\eta = log\mu$
- \$M = \$ diagonal matrix of link derivatives of η w.r.t. μ
  - So, $M = diag(1/\mu)$
- $W = (1/\mu)(1/\mu)^{-2} = \mu$

In R:

```
y <- gala$Species
mu <- y
M <- diag(1/mu)
eta <- log(mu)
z <- eta + M%*%(y-mu)
w <- mu
lmod <- lm(z ~ Area + Elevation + Nearest + Scruz + Adjacent, weights=w, gala)
coef(lmod)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz
##  3.5191545412 -0.0005298484  0.0031643557  0.0025188990 -0.0037899780
##      Adjacent
## -0.0006623523
```

```
coef(modpois)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz 
##  3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223 
##      Adjacent 
## -0.0006630311
```

The coefficients are remarkably close to the glm estimates.

d)

```
eta <- lmod$fit
mu <- exp(eta)
M <- diag(1/mu)
z <- eta + M%*%(y-mu)
w <- mu
lmod <- lm(z ~ gala$Area + gala$Elevation + 
       gala$Nearest + gala$Scruz + gala$Adjacent, weights = w)
coef(lmod)
```

```
##    (Intercept)      gala$Area gala$Elevation   gala$Nearest     gala$Scruz 
##   3.2102594447  -0.0005651969   0.0034606226   0.0077171134  -0.0052400871 
##  gala$Adjacent 
##  -0.0006604828
```

```
deviance <- 2*sum(y*log(y/mu) - (y-mu))
deviance
```

```
## [1] 828.0096
```

```
modpois$deviance
```

```
## [1] 716.8458
```

This calculated deviance is about 15% away from the deviance calculated in the glm function. This is fairly close.

e)

```
eta <- lmod$fit
mu <- exp(eta)
M <- diag(1/mu)
z <- eta + M%*%(y-mu)
w <- mu
lmod <- lm(z ~ gala$Area + gala$Elevation + 
       gala$Nearest + gala$Scruz + gala$Adjacent, weights = w)
coef(lmod)
```

```
##    (Intercept)      gala$Area gala$Elevation   gala$Nearest     gala$Scruz 
##   3.1562582546  -0.0005793855   0.0035379237   0.0087861184  -0.0056868875 
##  gala$Adjacent 
##  -0.0006630167 
```

```
coef(modpois)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz 
##  3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223 
##      Adjacent 
## -0.0006630311 
```

The coefficients are almost exactly the same as the glm fit. Differences may be chalked up to rounding errors in a real world application.

```
deviance <- 2*sum(y*log(y/mu) - (y-mu))
deviance
```

```
## [1] 719.4158
```

```
modpois$deviance
```

```
## [1] 716.8458
```

Deviance has come much closer to the value given by the glm fit.

f)

```
for (i in 1:3){
  eta <- lmod$fit
  mu <- exp(eta)
  M <- diag(1/mu)
  z <- eta + M%*%(y-mu)
  w <- mu
  lmod <- lm(z ~ gala$Area + gala$Elevation +
       gala$Nearest + gala$Scruz + gala$Adjacent, weights = w)
  coef(lmod)
  deviance <- 2*sum(y*log(y/mu) - (y-mu))
  cat(i,coef(lmod), deviance, "\n")
}
```

```
## 1 3.154809 -0.0005799422 0.003540591 0.008825509 -0.00570938 -0.0006630313 716.8488 
## 2 3.154808 -0.0005799429 0.003540594 0.008825572 -0.005709422 -0.0006630311 716.8458 
## 3 3.154808 -0.0005799429 0.003540594 0.008825572 -0.005709422 -0.0006630311 716.8458 
```

```
coef(modpois)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz
##  3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223
##      Adjacent
## -0.0006630311
```

After 2 times, deviance converged to the value given by the glm fit.

The final estimated coefficients are essentially exactly the same as the glm fit. There are a few differences in which decimal was chosen as the endpoint, but overall the coefficients are the same.

g)

```
summary(lmod)$coef[,2]/summary(lmod)$sigma
```

```
##    (Intercept)      gala$Area gala$Elevation    gala$Nearest     gala$Scruz
##   5.174955e-02   2.627299e-05   8.740709e-05   1.821261e-03   6.256214e-04
##  gala$Adjacent
##   2.932754e-05
```

```
summary(modpois)$coef[,c(1,2)]
```

```
##                  Estimate   Std. Error
## (Intercept)  3.1548078779 5.174952e-02
## Area        -0.0005799429 2.627298e-05
## Elevation    0.0035405940 8.740704e-05
## Nearest      0.0088255719 1.821260e-03
## Scruz       -0.0057094223 6.256200e-04
## Adjacent    -0.0006630311 2.932754e-05
```

The standard errors obtained are very similar or the same as the ones from the glm fit.

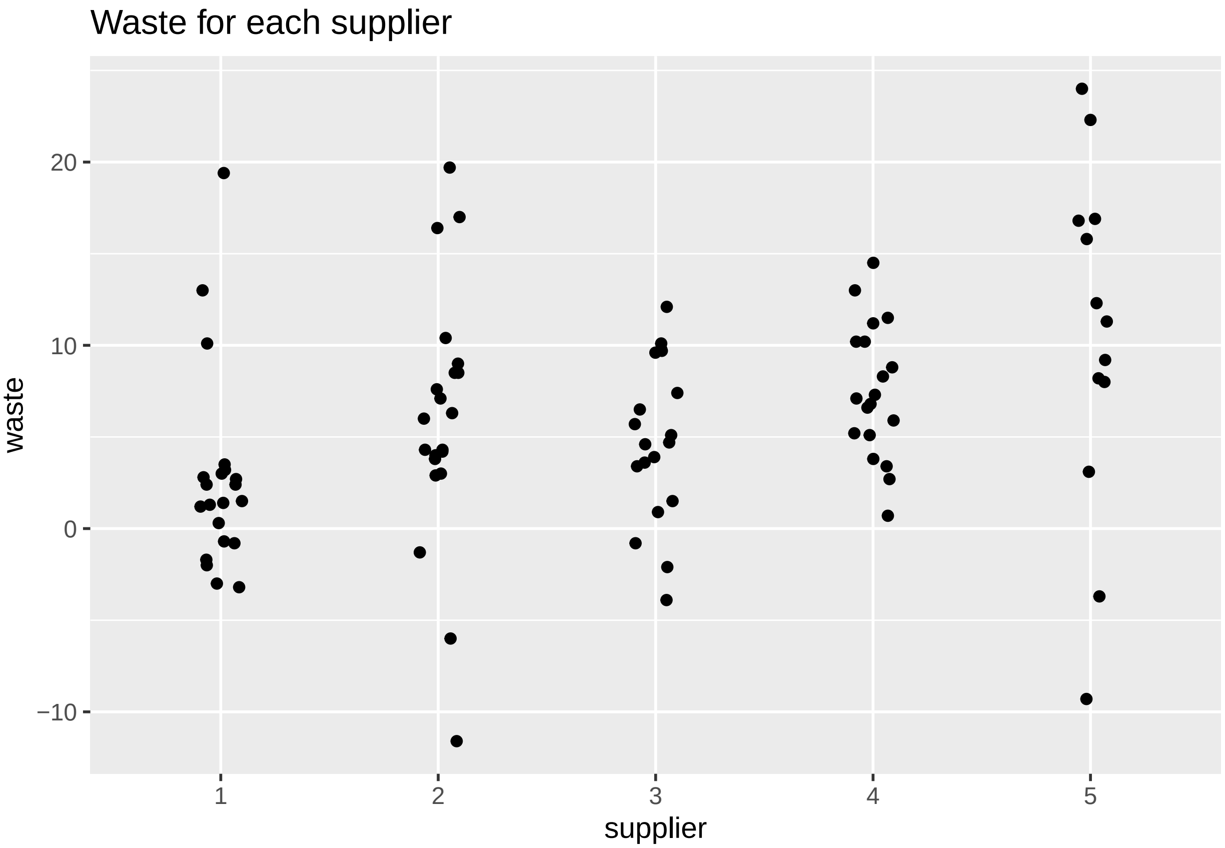
Question 3

```
library(faraway)
data(denim)
denim <- denim[-which(denim$waste == max(denim$waste)),] #removing 2 outliers
denim <- denim[-which(denim$waste == max(denim$waste)),]
str(denim)
```

```
## ’data.frame':    93 obs. of  2 variables:
##  $ waste   : num  1.2 16.4 12.1 11.5 24 10.1 -6 9.7 10.2 -3.7 ...
##  $ supplier: Factor w/ 5 levels "1","2","3","4",..: 1 2 3 4 5 1 2 3 4 5 ...
##  - attr(*, "na.action")= ’omit’ Named int [1:15] 70 75 80 85 90 95 98 99 100 103 ...
##   ..- attr(*, "names")= chr [1:15] "70" "75" "80" "85" ...
```

a)

```
suppressMessages(library(dplyr))
suppressMessages(library(ggplot2))
ggplot(denim, aes(x=supplier, y=waste))+
  geom_point(position = position_jitter(width=0.1, height=0.0)) +
  labs(title="Waste for each supplier")
```



From the plot above, we can see that the amount of waste for supplier 5 is the greatest of them all while having the widest range of waste. Suppliers 3 and 4 have a small range of waste produced. Supplier 2 is similar to supplier 5, but with a smaller minimum and a smaller maximum.

b)

```
op <- options(contrasts=c("contr.sum", "contr.poly"))
lmod <- aov(waste~supplier, denim)
summary(lmod)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)   
## supplier     4    545  136.16   3.657 0.00838 **
## Residuals   88   3277   37.24                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Under a null hypothesis that the supplier effect is 0 with a 0.05 significance level, we can reject the null hypothesis after observing a p-value under 0.05 (0.00838). THis means that we can determine that the supplier effect is significant

c)

```r
suppressMessages(library(lme4))
mmod <- lmer(waste~ 1 + (1|supplier), denim)
summary(mmod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: waste ~ 1 + (1 | supplier)
##    Data: denim
##
## REML criterion at convergence: 603.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.99119 -0.48597 -0.08981  0.49970  2.60002
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  supplier (Intercept)  5.718   2.391
##  Residual             37.292   6.107
## Number of obs: 93, groups:  supplier, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    6.155      1.246   4.938
```

The estimated standard deviations of the effects are 2.391 for supplier in the random effect and 1.246 for the intercept in the fixed effects.

d)

The correlation for observations within the same supplier is shown as:

$$\rho \equiv \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma^2}$$

according to the coursework.

In this case,

- σ_α^2 is 5.718 from the summary output
- σ^2 is 37.292 from the summary output

Thus the correlation within the same supplier should be:

$$\rho = 5.718/(5.718 + 37.292) = 0.1329458$$

e)

Checking if groups are balanced: they appear to be slightly unbalanced, with supplier 5 having noticeably less observations than the rest who are balanced with each other.

```
suppressMessages(library(plyr))
lapply(denim["supplier"], count)
```

```
## $supplier
##   x freq
## 1 1   21
## 2 2   21
## 3 3   19
## 4 4   19
## 5 5   13
```

The method below uses a bootstrap to calculate the p-value

```
nullmod <- lm(waste~1, denim)
proposed <- lmer(waste ~ 1 + (1|supplier), denim, REML=FALSE)
LRTstat <- as.numeric(2*(logLik(proposed) - logLik(nullmod)))

y <- simulate(nullmod)
LRTstat_star <- numeric(1000)
set.seed(123)
suppressMessages(for (i in 1:1000){
  y <- unlist(simulate(nullmod))
  bnull <- lm(y~1)
  balt <- lmer(y~1 + (1|supplier), denim, REML = FALSE)
  LRTstat_star[i] <- as.numeric(2*(logLik(balt) - logLik(bnull)))
})

mean(LRTstat_star > LRTstat) # p-value
```

```
## [1] 0.014
```

We can reasonably sure that our p-value is under 0.05. The p-value obtained through the bootstrap is not as strong as the one using just the fixed effects model.

Question 4

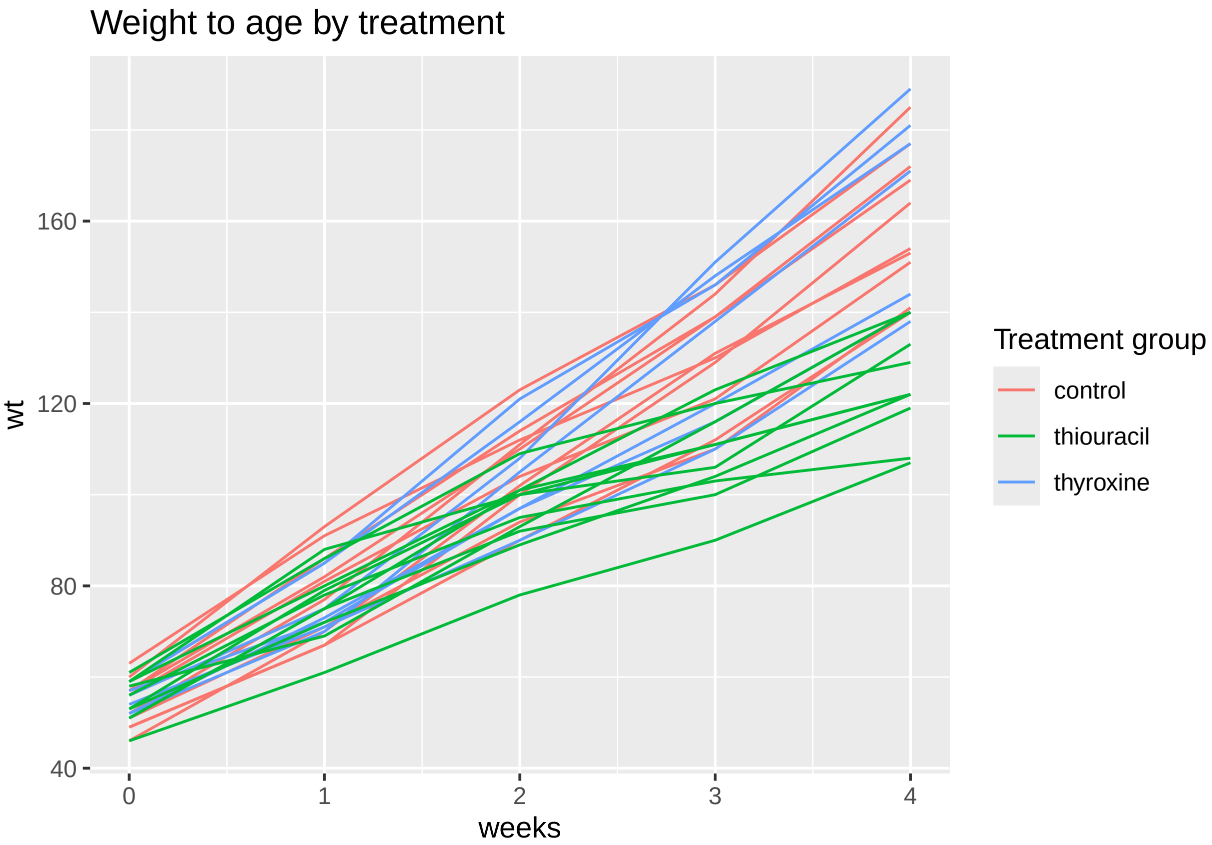
```
suppressMessages(library(faraway))
str(ratdrink)
```

```
## 'data.frame':    135 obs. of  4 variables:
##  $ wt     : num  57 86 114 139 172 60 93 123 146 177 ...
##  $ weeks  : int  0 1 2 3 4 0 1 2 3 4 ...
##  $ subject: Factor w/ 27 levels "1","2","3","4",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ treat  : Factor w/ 3 levels "control","thiouracil",..: 1 1 1 1 1 1 1 1 1 1 ...
```

a)

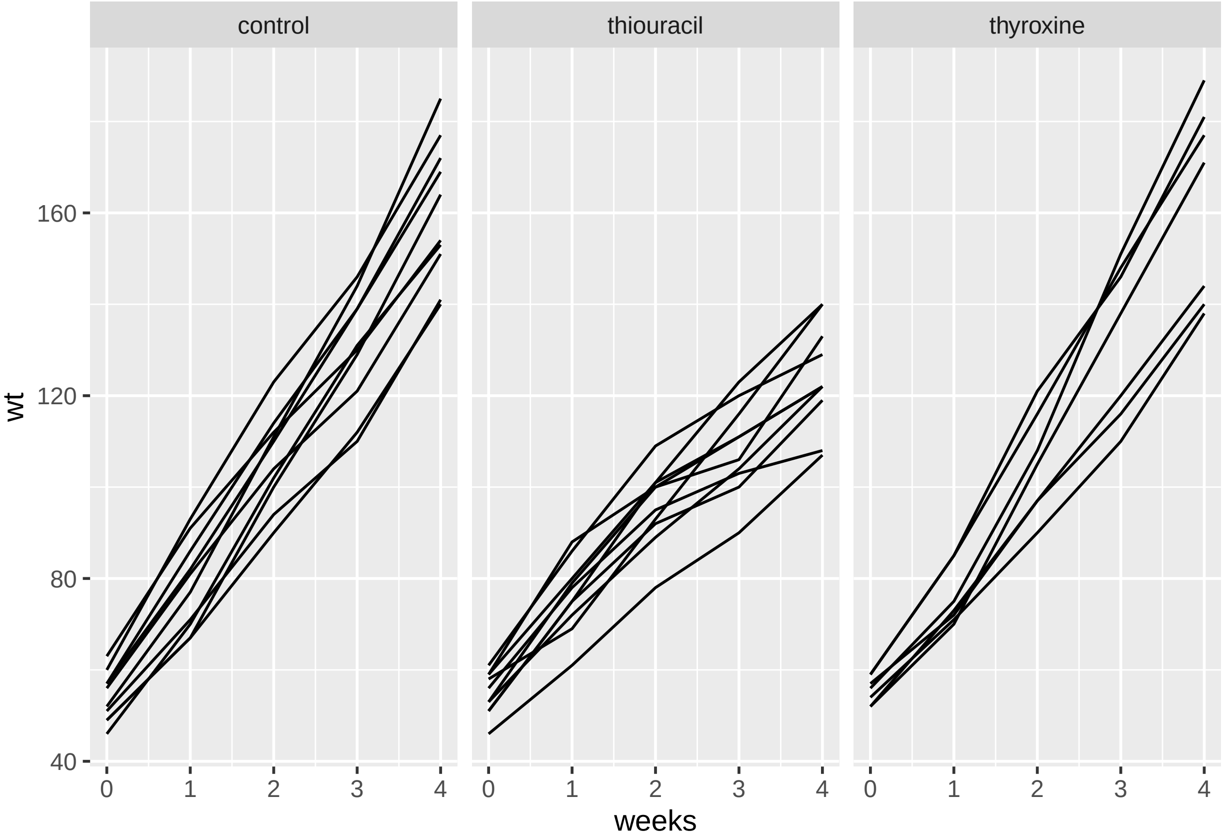
Single panel plot:

```
suppressMessages(library(dplyr))
suppressMessages(library(ggplot2))
ggplot(ratdrink, aes(x=weeks, y=wt, group=subject, color = treat)) +
  geom_line() +
  labs(title = "Weight to age by treatment", color = "Treatment group")
```



Three-panel plot:

```
ggplot(ratdrink, aes(x=weeks, y=wt, group=subject)) + geom_line() +
  facet_wrap(~treat)
```

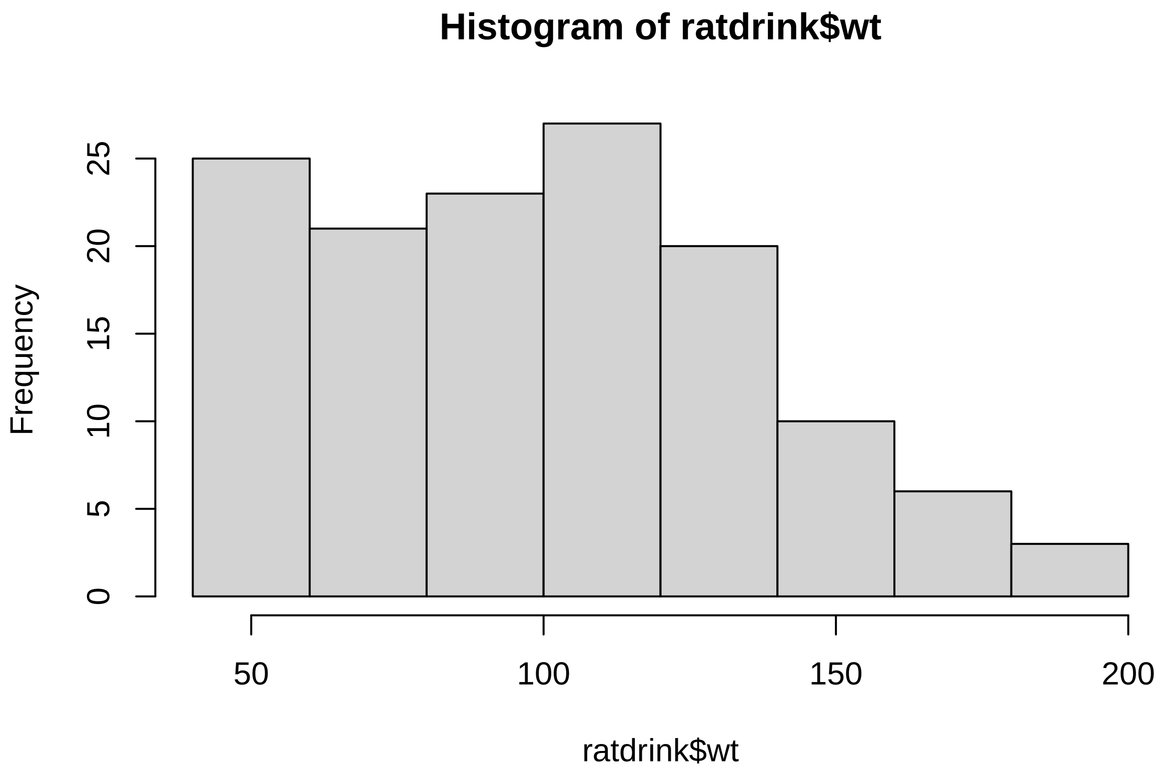


It appears that the group with thiouracil does not increase weight as much as the other groups. The group with thyroxine and the control group display a similar growth path, which may indicate that thyroxine does not affect the rats' weights.

b)

We show here that the weight variable is slightly right skewed, so we will correct by taking the log in our model

```
hist(ratdrink$wt)
```



```
suppressMessages(library(lme4))
set.seed(123)
ml <- lmer(log(wt) ~ weeks + weeks:treat + treat + (weeks|subject), ratdrink)
summary(ml)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(wt) ~ weeks + weeks:treat + treat + (weeks | subject)
##    Data: ratdrink
##
## REML criterion at convergence: -214.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.92720 -0.65956 -0.01151  0.65079  1.88194
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  subject  (Intercept) 0.0064579 0.08036
##           weeks       0.0003888 0.01972  -0.26
##  Residual             0.0056950 0.07546
## Number of obs: 135, groups:  subject, 27
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)   4.060625   0.019396 209.359
## weeks         0.245755   0.006042  40.674
## treat1       -0.010520   0.026558  -0.396
## treat2        0.018431   0.026558   0.694
## weeks:treat1  0.023099   0.008273   2.792
## weeks:treat2 -0.046622   0.008273  -5.635
##
## Correlation of Fixed Effects:
##              (Intr) weeks  treat1 treat2 wks:t1
## weeks        -0.506
## treat1       -0.091  0.046
## treat2       -0.091  0.046 -0.400
## weeks:tret1   0.046 -0.091 -0.506  0.202
## weeks:tret2   0.046 -0.091  0.202 -0.506 -0.400
```

i)

The fixed effect intercept term represents the average outcome when all covariates are at 0. This means the control group in this case. Thus a value of 4.0501046 for the fixed effect intercept represents the log-mean-weight of a rat in the control group at week 0. Exponentiating, we get a weight of 57.40346, which appears accurate according to our plots in **a)**

ii)

The interaction between thiouracil and week represents the log-weight change when the rats have thiouracil in their water vs when they are the control rats. For a value of -0.0697213, once we exponentiate it, we can see that being on thiouracil contributes to a $1 - 0.9326537 = 6.73463\%$ decrease in weight per week for the rats. This change may explain why the rats with thiouracil do not show as much growth as the other two treatments.

iii)

The intercept random effect SD represents the standard deviation of the random effect, which in this case, would be how much the log-weight of a supposed rat would deviate from the overall intercept. With an estimate of 0.0064579 and an SD of 0.08036, we can see that the average rat should not be too different from the overall intercept.

c)

To perform a hypothesis test for if the treatment is significant, we can use Kenward-Roger test to test the significance of the treatment term. I created a variety of models to test the treatments because I know that the presence of other predictors can change the significance of the one being tested.

```r
set.seed(123)
suppressWarnings(suppressMessages(library(pbkrtest)))
# Model with main and interaction effect
mmod <- lmer(log(wt)~ weeks + weeks:treat + treat +
               (weeks|subject), ratdrink, REML = FALSE)
# Model without interaction effect
mmodr <- lmer(log(wt)~ weeks + treat + (weeks|subject),
              ratdrink, REML = FALSE)
# Model without either main or interaction effect
mmodr2 <- lmer(log(wt) ~ weeks + (weeks|subject),
               ratdrink, REML = FALSE)
# Model without main effect
```

```
mmodr3 <- lmer(log(wt)~ weeks + weeks:treat +
               (weeks|subject), ratdrink, REML = FALSE)
```

We can see below that the p-value of the interaction is significant.

```
KRmodcomp(mmod, mmodr) # testing interaction
```

```
## large : log(wt) ~ weeks + treat + (weeks | subject) + weeks:treat
## small : log(wt) ~ weeks + treat + (weeks | subject)
##         stat    ndf    ddf F.scaling   p.value
## Ftest 16.049  2.000 24.000         1 3.759e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Meanwhile, the below 2 tests, done on models that have the treatment main effect vs models that do not have the main effect, show that the treatment main effect is not significant.

```
KRmodcomp(mmodr, mmodr2) # testing main effect on no interaction
```

```
## large : log(wt) ~ weeks + treat + (weeks | subject)
## small : log(wt) ~ weeks + (weeks | subject)
##          stat     ndf     ddf F.scaling p.value
## Ftest  2.9014  2.0000 24.0000         1 0.07438 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

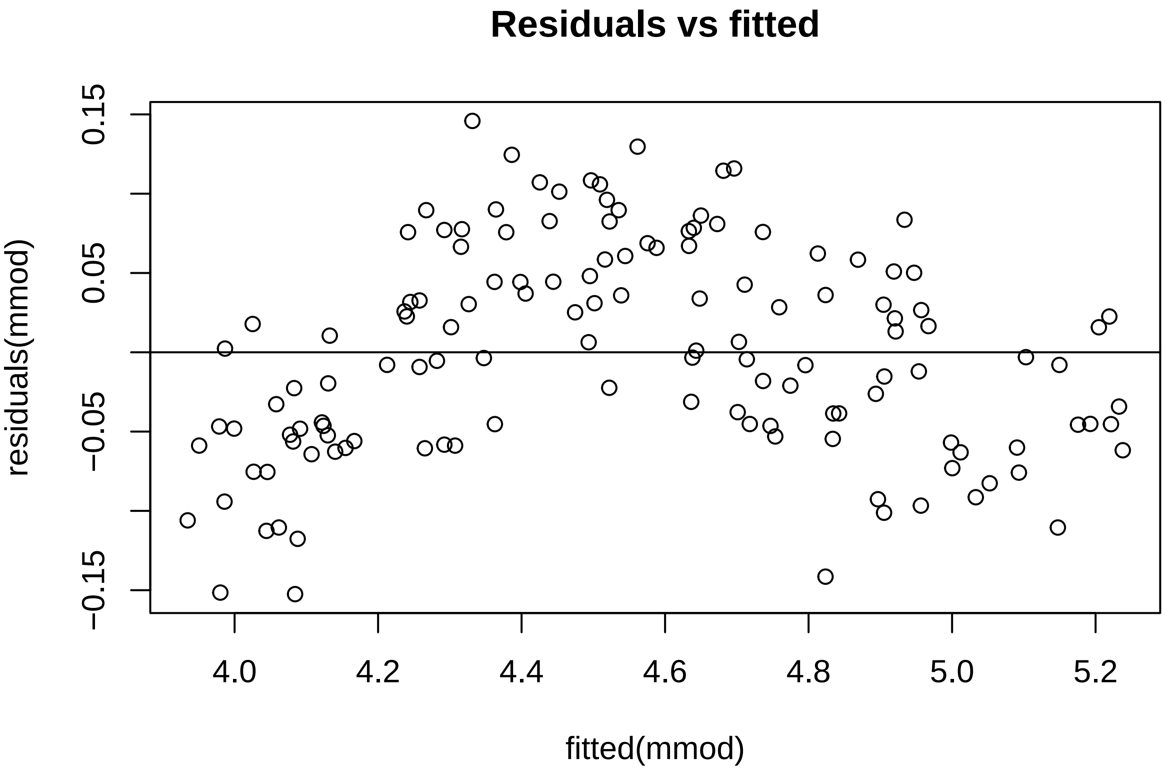
```
KRmodcomp(mmod, mmodr3) # testing main effect on interaction
```

```
## large : log(wt) ~ weeks + treat + (weeks | subject) + weeks:treat
## small : log(wt) ~ weeks + weeks:treat + (weeks | subject)
##          stat     ndf     ddf F.scaling p.value
## Ftest  0.2492  2.0000 24.0000         1  0.7814
```

d)

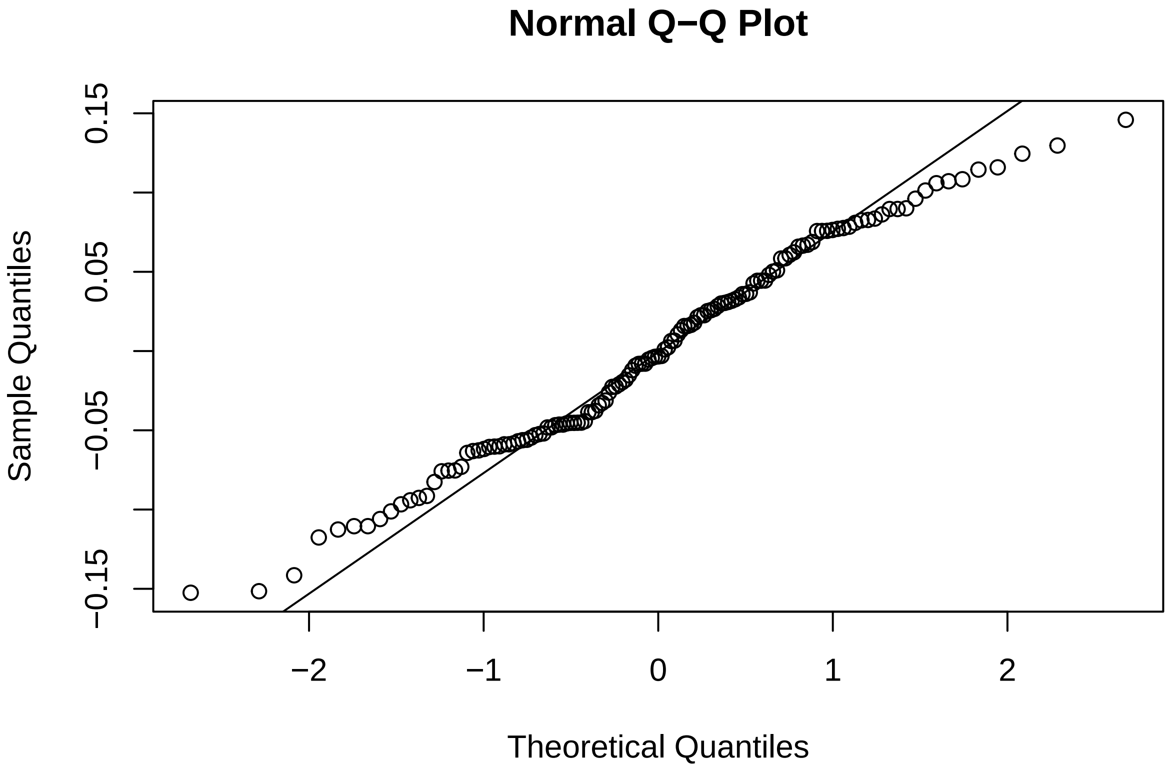
Residuals vs fitted: we should see that the residuals vs fitted plot has a constant variance of point across the entire plot, but this is not the case. We see an almost parabolic relationship in the points, which suggests that the variance of the data is not independent and identically distributed as we have assumed by using a mixed effects model.

```
plot(fitted(mmod), residuals(mmod), main = "Residuals vs fitted")
# residuals vs fitted
abline(h=0)
```



QQ residuals plot: In the tail ends of the plot, we see a deviance from the normality assumption, again suggesting that there is a problem with homoscedasticity in the model.

```
qqnorm(residuals(mmod))
qqline(residuals(mmod))
```



e)

```
set.seed(123)
suppressMessages(confint(mmod, method="boot",oldNames = FALSE))
```

```
##                                         2.5 %      97.5 %
## sd_(Intercept)|subject         0.0340548479  0.09846617
## cor_weeks.(Intercept)|subject -1.0000000000  1.00000000
## sd_weeks|subject               0.0008096767  0.02549821
## sigma                          0.0637683949  0.08653987
## (Intercept)                    4.0236372553  4.10137283
## weeks                          0.2346274068  0.25638448
## treat1                        -0.0629391218  0.04601433
## treat2                        -0.0357678199  0.07003347
## weeks:treat1                   0.0056758046  0.03973239
## weeks:treat2                  -0.0604377105 -0.03221399
```

We should remove the random effect between weeks and subject since there is so much correlation, there is no other random effect term that we can remove since they do not include 0 in the interval.

Regarding the thyroxine group, since its confidence interval contains 0, this implies that it is not significantly different from the control group. We saw this behaviour in our plots in **a)**

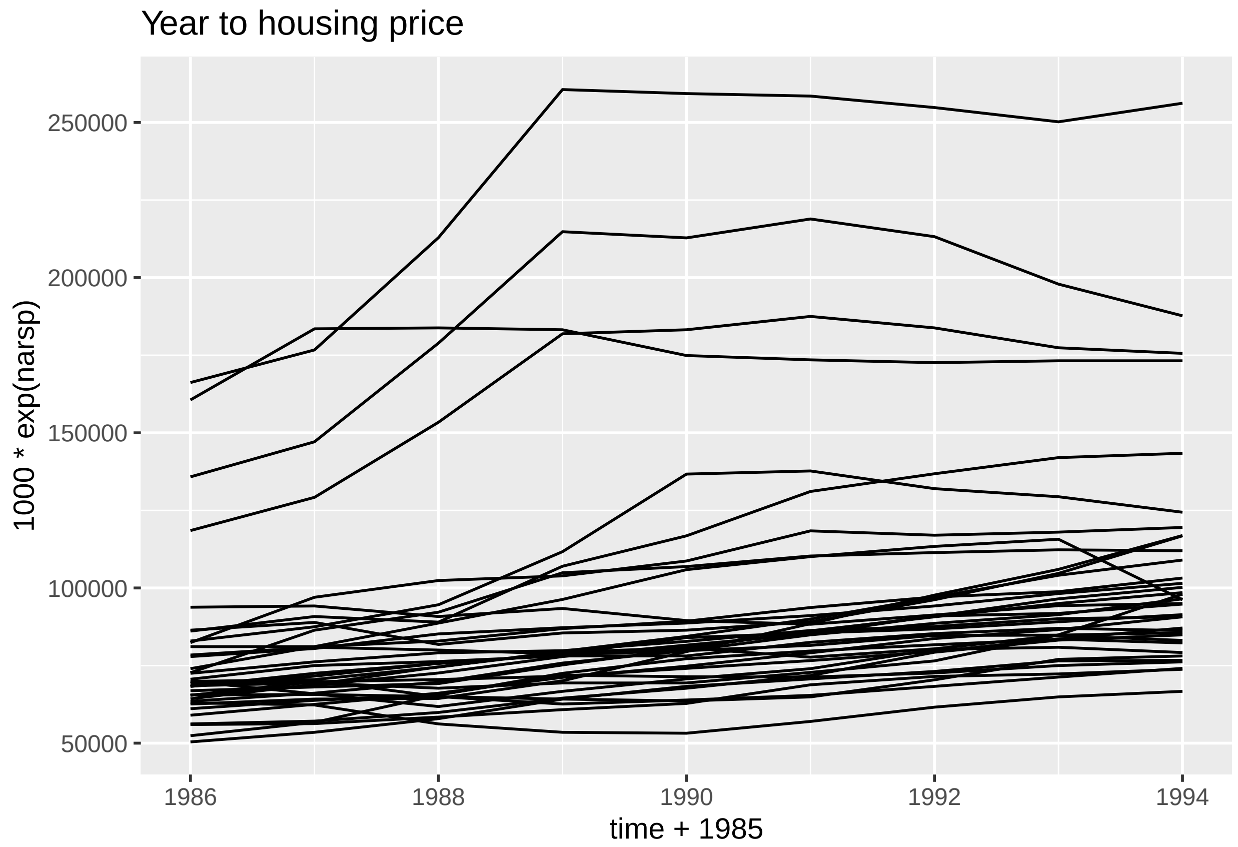
Question 5

```
suppressMessages(library(faraway))
str(hprice)
```

```
## 'data.frame':    324 obs. of  8 variables:
##  $ narsp  : num  4.22 4.27 4.33 4.36 4.39 ...
##  $ ypc    : int  13585 14296 15413 16490 17634 18210 17958 18659 19360 15354 ...
##  $ perypc : num  6.47 5.23 7.81 6.99 6.94 ...
##  $ regtest: int  20 20 20 20 20 20 20 20 20 18 ...
##  $ rcdum  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ajwtr  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ msa    : Factor w/ 36 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ time   : int  1 2 3 4 5 6 7 8 9 1 ...
```

a)

```
suppressMessages(library(dplyr))
suppressMessages(library(ggplot2))
ggplot(hprice, aes(x=time+1985, y=1000*exp(narsp), group=msa)) +
  geom_line() +
  labs(title = "Year to housing price")
```



The plot shows that prices rose throughout the whole time period, saw a greater increase in 1989, then slowed the increase rate until 1994 to a more normal average increase rate.

b)

Remember that the `narsp` variable is already the natural log average sale price in thousands of dollars.

```
suppressMessages(library(lme4))
model <- lm(narsp ~ ypc + perypc + regtest + rcdum + ajwtr + time, hprice)
summary(model)
```

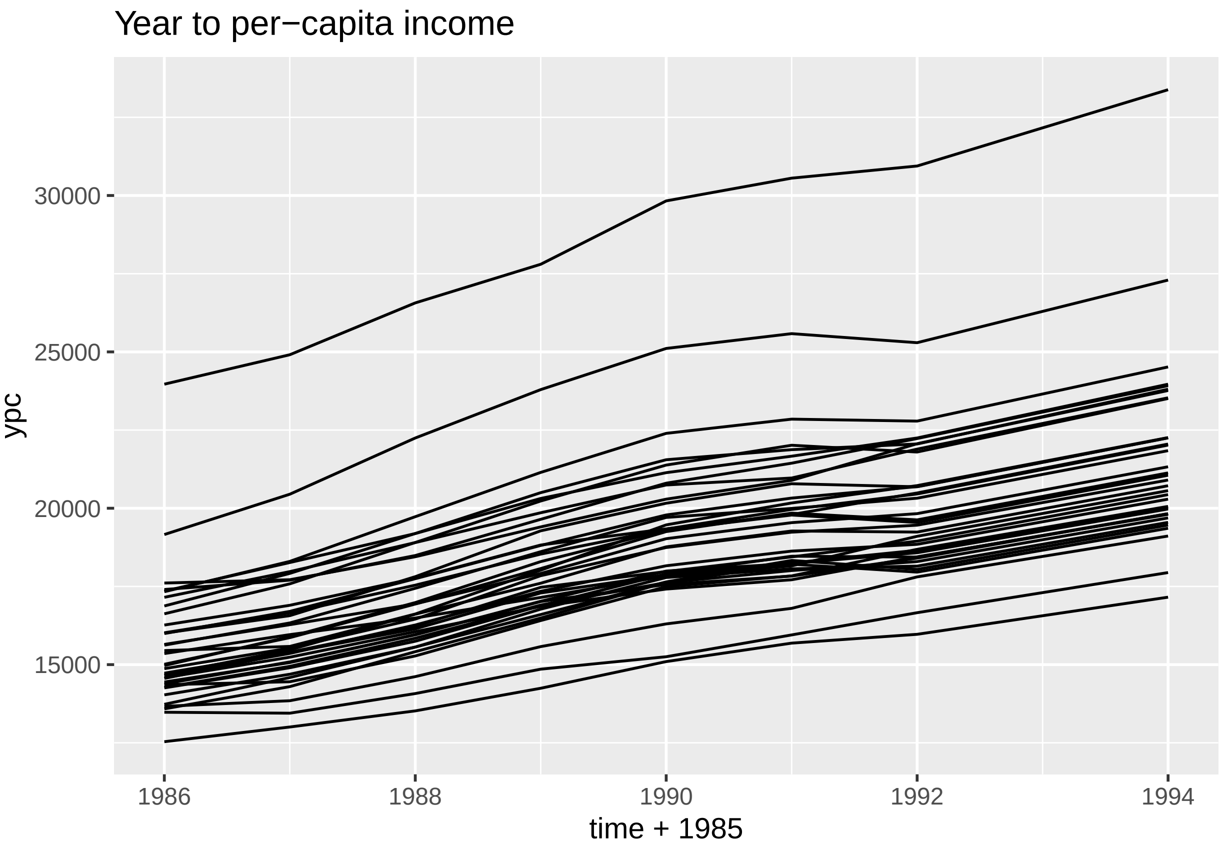
```
##
## Call:
## lm(formula = narsp ~ ypc + perypc + regtest + rcdum + ajwtr +
##     time, data = hprice)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31386 -0.10810 -0.01525  0.08547  0.55594
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.765e+00  9.398e-02  29.425  < 2e-16 ***
## ypc          7.029e-05  4.358e-06  16.128  < 2e-16 ***
## perypc      -1.372e-02  5.074e-03  -2.704 0.007216 **
## regtest      2.954e-02  3.103e-03   9.520  < 2e-16 ***
## rcdum1      -7.440e-02  1.618e-02  -4.599 6.15e-06 ***
## ajwtr1      -1.797e-02  1.001e-02  -1.796 0.073482 .
## time        -1.767e-02  5.128e-03  -3.445 0.000647 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1651 on 317 degrees of freedom
## Multiple R-squared:  0.7572, Adjusted R-squared:  0.7526
## F-statistic: 164.7 on 6 and 317 DF,  p-value: < 2.2e-16
```

All terms except for `ajwtr1` are significant, which suggests that being adjacent to a coastline is not an important factor in predicting housing prices.

The coefficient for time is the percent change in log-house prices per 1 unit increase in time. In this case it is a 1.767% decrease for every 1 unit increase in time (years).

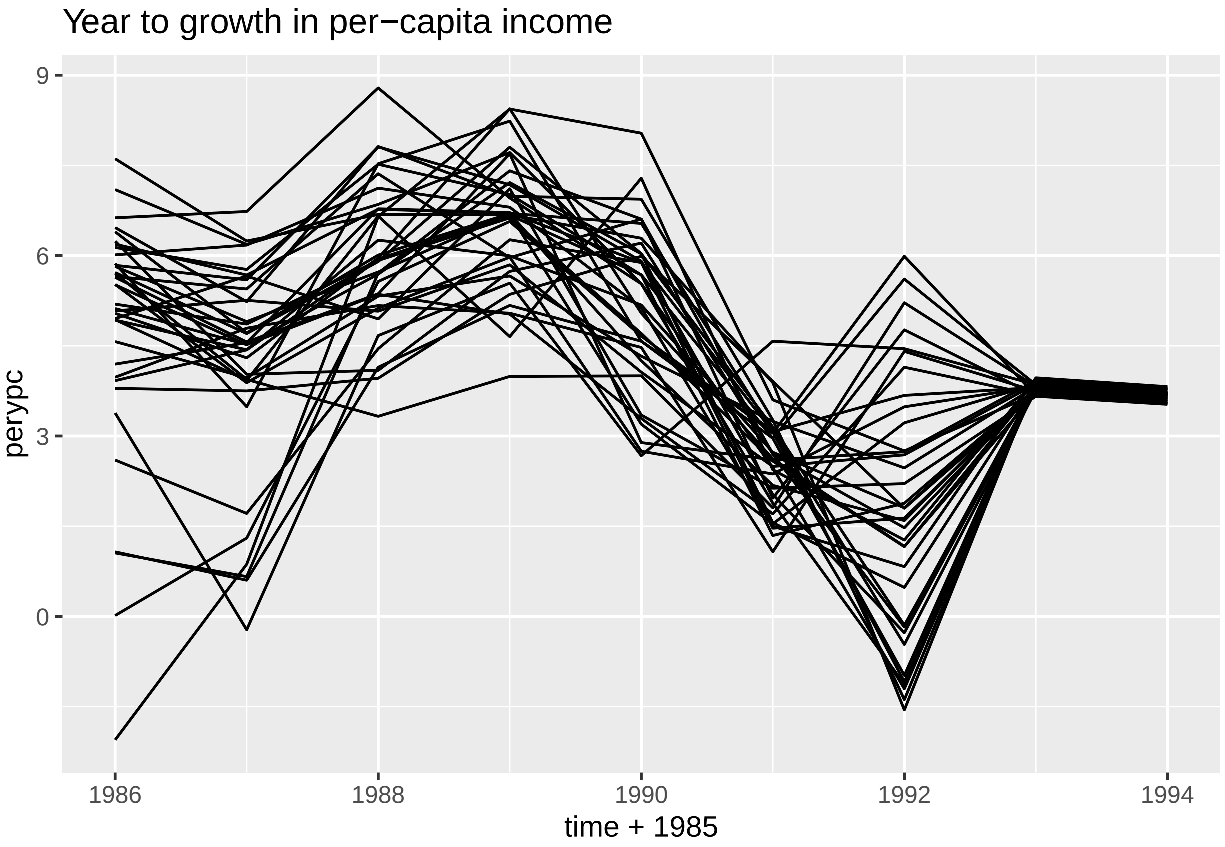
c)

```
ggplot(hprice, aes(x=time+1985, y=ypc, group=msa)) + geom_line() +
  labs(title = "Year to per-capita income")
```



Generally, per-capita income increased linearly over the years. There is a drop in 1991 to 1992 that appears like a hump in the graph, but the rates continued on linearly after that.

```
ggplot(hprice, aes(x=time+1985, y=perypc, group=msa)) + geom_line() +
  labs(title = "Year to growth in per-capita income")
```



In this plot, we can see that the nature of the growth in per-capita income is an indicator for the per-capita income. This makes sense since we are looking at the derivative of per-capita income with respect to time. This makes the movements in the first graph more drastic, as we can see the drop in 1991-1992 like I mentioned earlier more clearly.

d)

```
year1 <- subset(hprice, time == 1, select = c(msa, ypc))
names(year1)[names(year1) == "ypc"] <- "ypc_year1"
hprice <- merge(hprice, year1, by = "msa", all.x = TRUE)
```

```
# should take the log of ypc_year1 to not overpower the other predictors
newmodel <- lm(narsp ~ log(ypc_year1) + perypc + regtest +
                 rcdum + ajwtr + time, hprice)
summary(newmodel)
```

```
##
## Call:
## lm(formula = narsp ~ log(ypc_year1) + perypc + regtest + rcdum +
##     ajwtr + time, data = hprice)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31161 -0.10564 -0.01940  0.08439  0.52039
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -10.867345   0.838965 -12.953  < 2e-16 ***
## log(ypc_year1)   1.516190   0.087180  17.391  < 2e-16 ***
## perypc          -0.008317   0.004890  -1.701   0.0899 .
## regtest          0.031233   0.002975  10.500  < 2e-16 ***
## rcdum1          -0.080434   0.015433  -5.212 3.38e-07 ***
## ajwtr1          -0.018127   0.009642  -1.880   0.0610 .
## time             0.037051   0.003723   9.951  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1593 on 317 degrees of freedom
## Multiple R-squared:  0.7738, Adjusted R-squared:  0.7695
## F-statistic: 180.7 on 6 and 317 DF,  p-value: < 2.2e-16
```

```r
summary(model)
```

```
##
## Call:
## lm(formula = narsp ~ ypc + perypc + regtest + rcdum + ajwtr +
##     time, data = hprice)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31386 -0.10810 -0.01525  0.08547  0.55594
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.765e+00  9.398e-02  29.425  < 2e-16 ***
## ypc          7.029e-05  4.358e-06  16.128  < 2e-16 ***
## perypc      -1.372e-02  5.074e-03  -2.704 0.007216 **
## regtest      2.954e-02  3.103e-03   9.520  < 2e-16 ***
## rcdum1      -7.440e-02  1.618e-02  -4.599 6.15e-06 ***
## ajwtr1      -1.797e-02  1.001e-02  -1.796 0.073482 .
## time        -1.767e-02  5.128e-03  -3.445 0.000647 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1651 on 317 degrees of freedom
## Multiple R-squared:  0.7572, Adjusted R-squared:  0.7526
## F-statistic: 164.7 on 6 and 317 DF,  p-value: < 2.2e-16
```

In the new model,

- The intercept decreased
- The coefficient for per-capita income increased
- The coefficient for percentage growth in per-capita income decreased and is not significant
- Regtest increased

- rcdum1 increased
- ajwtr1 increased and is not significant
- Time became positive
- R^2 and adjusted R^2 grew, suggesting that the new predictors are more effective at predicting the log-price of a home
- Residual standard error decreased

These effects make sense because if your income stays the same, then the value of the other variables becomes greater for the most part in predicting the sale price of a house.

e)

```r
set.seed(123)
lmermodel <- lmer(narsp ~ log(ypc_year1) + perypc + regtest +
                    rcdum + ajwtr + time + (1|msa), hprice)
summary(lmermodel)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: narsp ~ log(ypc_year1) + perypc + regtest + rcdum + ajwtr + time +
##     (1 | msa)
##    Data: hprice
##
## REML criterion at convergence: -599.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3652 -0.5959 -0.0240  0.5815  2.8581
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  msa      (Intercept) 0.022646 0.15049
##  Residual             0.005451 0.07383
## Number of obs: 324, groups:  msa, 36
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)    -10.861469   2.406604  -4.513
## log(ypc_year1)   1.516052   0.250325   6.056
## perypc          -0.009152   0.002298  -3.983
## regtest          0.031248   0.008538   3.660
## rcdum1          -0.080501   0.044301  -1.817
## ajwtr1          -0.018120   0.027686  -0.655
## time             0.036803   0.001729  21.282
##
## Correlation of Fixed Effects:
##               (Intr) lg(_1) perypc regtst rcdum1 ajwtr1
## lg(ypc_yr1) -0.997
## perypc      -0.007  0.002
## regtest      0.074 -0.151 -0.005
## rcdum1      -0.348  0.310  0.004  0.325
## ajwtr1      -0.189  0.185 -0.001  0.044 -0.182
## time        -0.006  0.001  0.395 -0.002  0.002  0.000
```

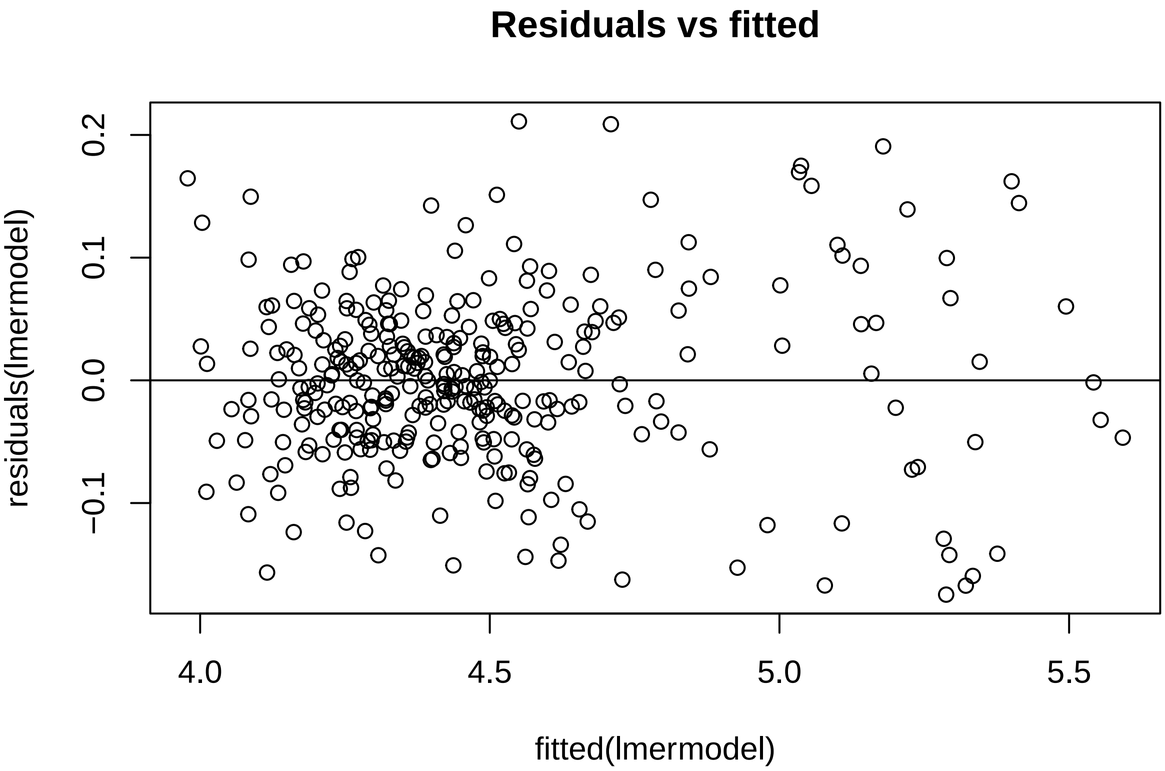
Each MSA may affect the housing prices differently, as we saw in part **a)**, although the overall trend was similar, some MSAs had remarkably higher housing prices than others.

The coefficient of time is how much the log-price of a home changes for every 1 year. In our model, this means that for every year, the log-price of the house price should increase by 0.036803%, or translated to the correct units, the house should increase in value by a factor of $exp(0.036803) = 1.037489$, or by 3.7489%.

f)

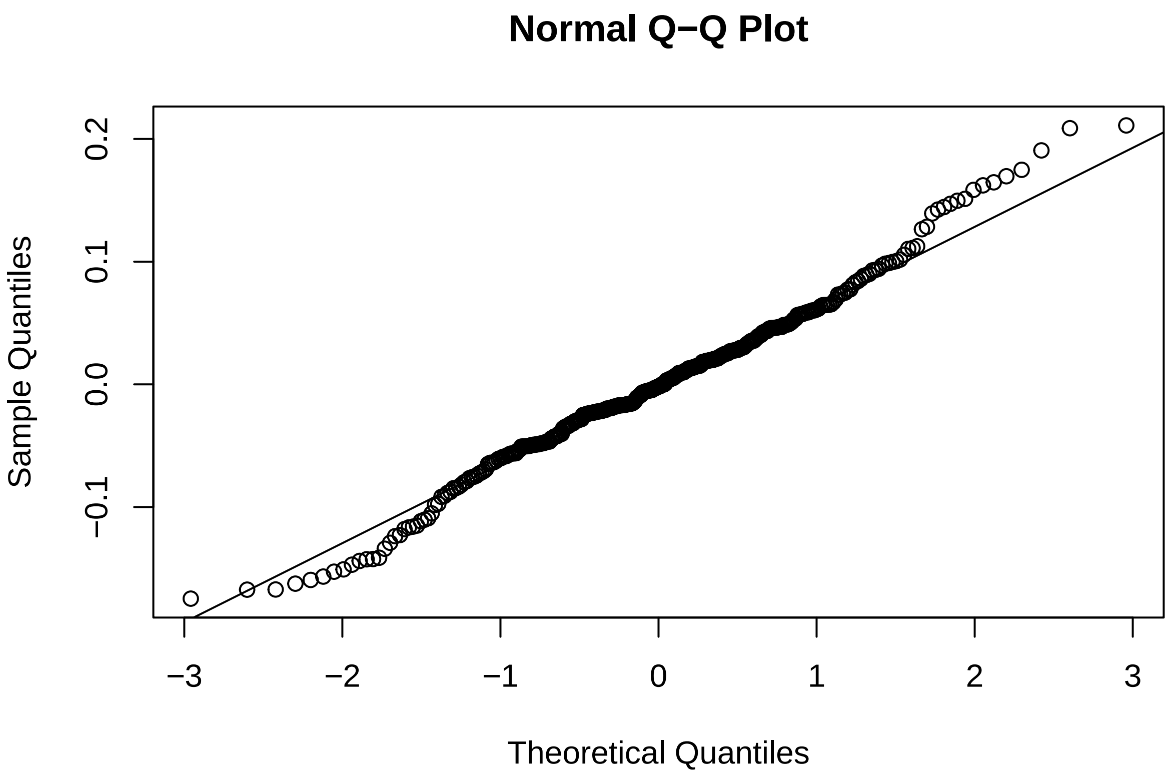
The fitted vs residuals plot shows that there might be a bit of clustering in the lower end of the model, but overall the variance appears to be consistent along the whole plot, confirming our assumption the data is normal.

```
plot(fitted(lmermodel), residuals(lmermodel),
     main = "Residuals vs fitted") # residuals vs fitted
abline(h=0)
```



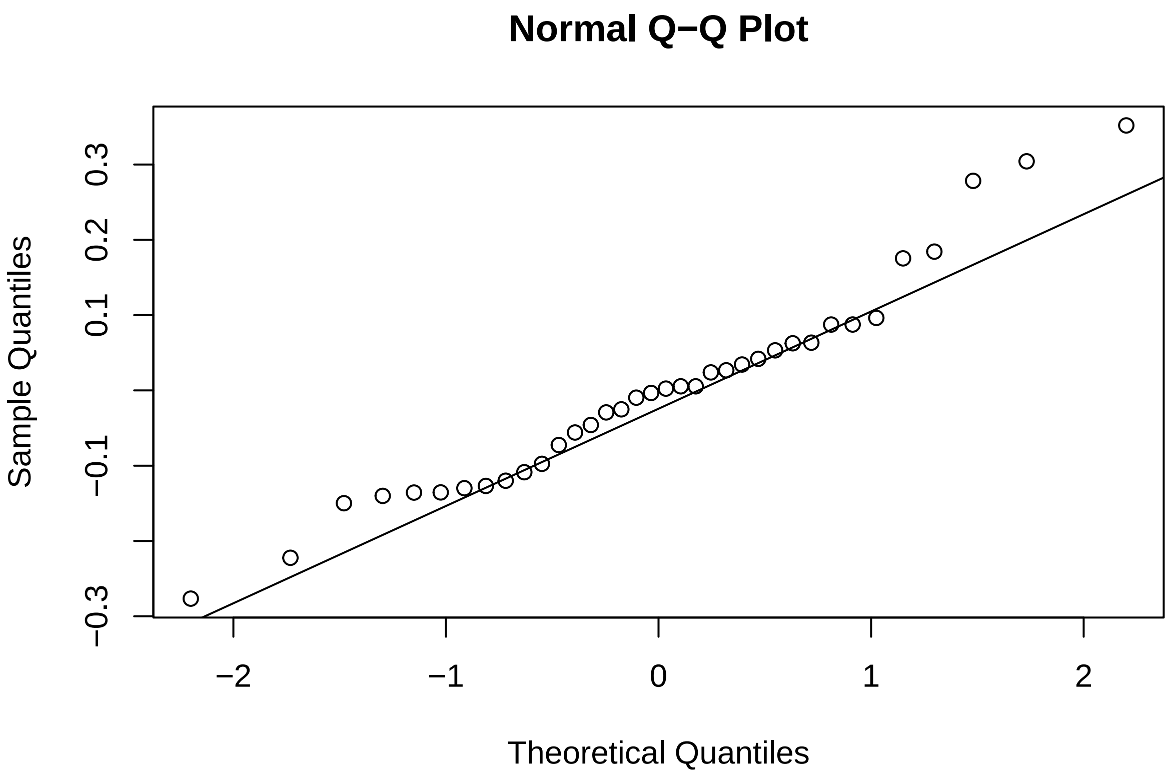
The residuals QQ plot looks good throughout most of the model, except in the tail ends where there is a slight deviance. Fortunately, the pattern does not completely deviate and instead returns to the normal line.

```
qqnorm(residuals(lmermodel))
qqline(residuals(lmermodel))
```



The QQ plot of the random effects does not hold the normality assumption as well as the residuals, which may imply that there is a homoscedasticity problem with the random effect model

```r
random <- ranef(lmermodel)$msa
qqnorm(random[,1])
qqline(random[,1])
```



g)

```
nullmod <- lmer(narsp ~ log(ypc_year1) + perypc + regtest + time +
                  (1|msa), hprice, REML=FALSE)
proposed <- lmer(narsp ~ log(ypc_year1) + perypc + regtest + rcdum +
                   ajwtr + time + (1|msa), hprice, REML=FALSE)
LRTstat <- as.numeric(2*(logLik(proposed) - logLik(nullmod)))

LRTstat_star <- numeric(1000)
set.seed(123)
suppressWarnings(suppressMessages(for (i in 1:1000){
  y <- unlist(simulate(nullmod))
  bnull <- lmer(y ~ ypc_year1 + perypc + regtest + time +
                  (1|msa), hprice, REML=FALSE)
  balt <- lmer(y ~ ypc_year1 + perypc + regtest + rcdum +
                 ajwtr + time + (1|msa), hprice, REML=FALSE)
  LRTstat_star[i] <- as.numeric(2*(logLik(balt) - logLik(bnull)))
}))

mean(LRTstat_star > LRTstat) # p-value
```

```
## [1] 0.145
```

According to the p-value of over 14% (0.145), the predictors for adjacent to water and rent control are not significant. This means that the reduction in predictors can be supported.