

Modelldiagnosztika

Zoltan Kekecs

November 26, 2021

Contents

1	Absztrakt	2
2	Adat kezelés és leíró statisztikák	2
2.1	Csomagok betöltése	2
2.2	Saját függvények	2
2.3	A King County, USA-beli ingatlanárakat tartalmazó adatsor beolvasása	3
2.4	Az adatsor megtekintése	3
3	Modell diagnosztika	4
3.1	Modell előállítás	5
4	Kiugró adatok kezelése	5
4.1	Szélsőséges esetek azonosítása	5
4.2	Jelentős hatású kiugró értékek azonosítása	6
5	A lineáris regresszió előfeltételei	8
5.1	Normalitás	9
5.2	Linearitás	14
5.3	Homoszkedaszticitás	16
5.4	A Multikollinearitás tesztelése	23

1 Absztrakt

Ebben a gyakorlatban arra térünk ki, hogyan elmemorizhatjuk hogy a lineáris regresszió előfeltetelei teljesülnek-e a modellünkre, milyen következményei vannak ha sérülnek ezek az előfeltételek és mi a teendő ilyenkor.

2 Adat kezelés és leíró statisztikák

2.1 Csomagok betöltése

Ennek a gyakorlatnak a során az alábbi csomagokat fogjuk használni:

```
library(psych) # for describe
library(car)   # for residualPlots, vif, pairs.panels, ncvTest
library(lmtest) # bptest
library(sandwich) # for coeftest vcovHC estimator
library(boot)  # for bootstrapping
library(lmboot) # for wild bootstrapping
library(tidyverse) # for tidy code
```

2.2 Saját függvények

Az ora során használunk majd saját függvényeket, melyek nem szerepelnek a fenti package-ekben. Ezeket töltsd be most hogy a későbbi kodok rendben lefussanak.

A bootstrapped confidencia intervallumok meghatározásához az alábbi saját függvényeket alkalmazzuk majd:

```
# function to obtain regression coefficients source:
# https://www.statmethods.net/advstats/bootstrapping.html
bs_to_boot <- function(model, data, indices) {
  d <- data[indices, ] # allows boot to select sample
  fit <- lm(formula(model), data = d)
  return(coef(fit))
}

# function to obtain adjusted R^2 source:
# https://www.statmethods.net/advstats/bootstrapping.html
# (partially modified)
adjR2_to_boot <- function(model, data, indices) {
  d <- data[indices, ] # allows boot to select sample
  fit <- lm(formula(model), data = d)
  return(summary(fit)$adj.r.squared)
}

# Computing the bootstrap BCa (bias-corrected and
# accelerated) bootstrap confidence intervals by Efron
# (1987) This is useful if there is bias or skew in the
# residuals.

confint.boot <- function(model, data = NULL, R = 1000) {
  if (is.null(data)) {
    data = eval(parse(text = as.character(model$call[3])))
  }
  boot.ci_output_table = as.data.frame(matrix(NA, nrow = length(coef(model)),
    ncol = 2))
```

```

row.names(boot.ci_output_table) = names(coef(model))
names(boot.ci_output_table) = c("boot 2.5 %", "boot 97.5 %")
results.boot = results <- boot(data = data, statistic = bs_to_boot,
  R = 1000, model = model)

for (i in 1:length(coef(model))) {
  boot.ci_output_table[i, ] = unlist(unlist(boot.ci(results.boot,
    type = "bca", index = i))[c("bca4", "bca5")]))
}

return(boot.ci_output_table)
}

# Computing the bootstrapped confidence interval for a
# linear model using wild bottstrapping as described by Wu
# (1986) <doi:10.1214/aos/1176350142> requires the lmbot
# package

wild.boot.confint <- function(model, data = NULL, B = 1000) {
  if (is.null(data)) {
    data = eval(parse(text = as.character(model$call[3])))
  }

  wild_boot_estimates = wild.boot(formula(model), data = data,
    B = B)

  result = t(apply(wild_boot_estimates[[1]], 2, function(x) quantile(x,
    probs = c(0.025, 0.975))))

  return(result)
}

```

2.3 A King County, USA-beli ingatlanárakat tartalmazó adatsor beolvasása

Ebben a gyakorlatban a különböző ingatlanok árának meghatározását tuzzük ki célul.

A Kaggle-bol származó adatsort fogjuk használni, amely tartalmazza az ingatlan árakat és az ezeket potenciálisan befolyásoló egyéb tényezok értékeit. Adatsorunk a King County, USA (Seattle és környéke)-beli ingatlanárakat tartalmazza, beleértve Seattle-t is. Az adatokat 2014 és 2015 Májusa között vették fel. További információ az adatsorról az alábbi linken: <https://www.kaggle.com/harlfoxem/housesalesprediction>

Mi az adatsornak csak egy részét fogjuk használni, összesen $N = 200$ ingatlant vizsgálva.

Az adatok az alábbi kód futtatásával olvashatóak be:

```
data_house = read.csv("https://bit.ly/2DpwK0r")
```

2.4 Az adatsor megtekintése

Fontos, hogy az elemzést mindig az adatsor megismerésével, és az esetleges ellentmondások javításával kezdjük.

A következő kódrészletben atváltjuk az USA dollar-t millió forint mertekegységre, az alapterület mértékegységét az eredeti négyzetlábról négyzetméterre alakítjuk, illetve a has_basement változót is megnevezzük mint faktort.

```
data_house %>%
  summary()
```

```
##           id           date           price           bedrooms
## Min.      :1.600e+07   Length:200       Min.       : 153503   Min.       :1.00
## 1st Qu.:1.885e+09     Class :character   1st Qu.: 299250   1st Qu.:3.00
## Median :3.521e+09     Mode  :character   Median : 425000   Median :3.00
## Mean     :4.113e+09                                Mean     : 453611   Mean     :2.76
## 3rd Qu.:6.424e+09                                3rd Qu.: 550000   3rd Qu.:3.00
## Max.     :9.819e+09                                Max.     :1770000   Max.     :3.00
##   bathrooms   sqft_living   sqft_lot   floors   waterfront
## Min.      :0.75   Min.      : 590   Min.      : 914   Min.      :1.000   Min.      :0.000
## 1st Qu.:1.00   1st Qu.:1240   1st Qu.: 4709   1st Qu.:1.000   1st Qu.:0.000
## Median :1.75   Median :1620   Median : 7270   Median :1.000   Median :0.000
## Mean     :1.85   Mean     :1728   Mean     :12985   Mean     :1.472   Mean     :0.005
## 3rd Qu.:2.50   3rd Qu.:1985   3rd Qu.:10187   3rd Qu.:2.000   3rd Qu.:0.000
## Max.     :3.50   Max.     :4380   Max.     :217800   Max.     :3.000   Max.     :1.000
##      view      condition      grade      sqft_above sqft_basement
## Min.      :0.000   Min.      :3.00   Min.      : 5.00   Min.      : 590   Min.      : 0.0
## 1st Qu.:0.000   1st Qu.:3.00   1st Qu.: 7.00   1st Qu.:1090   1st Qu.: 0.0
## Median :0.000   Median :3.00   Median : 7.00   Median :1375   Median : 0.0
## Mean     :0.145   Mean     :3.42   Mean     : 7.36   Mean     :1544   Mean     :184.1
## 3rd Qu.:0.000   3rd Qu.:4.00   3rd Qu.: 8.00   3rd Qu.:1862   3rd Qu.:315.0
## Max.     :4.000   Max.     :5.00   Max.     :11.00   Max.     :4190   Max.     :1600.0
##   yr_built   yr_renovated   zipcode   lat
## Min.      :1900   Min.      : 0.00   Min.      :98001   Min.      :47.18
## 1st Qu.:1946   1st Qu.: 0.00   1st Qu.:98033   1st Qu.:47.49
## Median :1968   Median : 0.00   Median :98065   Median :47.58
## Mean     :1968   Mean     : 79.98   Mean     :98078   Mean     :47.57
## 3rd Qu.:1993   3rd Qu.: 0.00   3rd Qu.:98117   3rd Qu.:47.68
## Max.     :2015   Max.     :2014.00   Max.     :98199   Max.     :47.78
##      long      sqft_living15   sqft_lot15   has_basement
## Min.      : -122.5   Min.      : 740   Min.      : 914   Length:200
## 1st Qu.: -122.3   1st Qu.:1438   1st Qu.: 5000   Class :character
## Median : -122.2   Median :1715   Median : 7222   Mode  :character
## Mean     : -122.2   Mean     :1793   Mean     :11225
## 3rd Qu.: -122.1   3rd Qu.:2072   3rd Qu.:10028
## Max.     : -121.7   Max.     :3650   Max.     :208652
```

```
data_house = data_house %>%
  mutate(price_mill_HUF = (price * 293.77)/1e+06, sqm_living = sqft_living *
    0.09290304, sqm_lot = sqft_lot * 0.09290304, sqm_above = sqft_above *
    0.09290304, sqm_basement = sqft_basement * 0.09290304,
    sqm_living15 = sqft_living15 * 0.09290304, sqm_lot15 = sqft_lot15 *
    0.09290304)
```

3 Modell diagnosztika

Valahányszor egy modellt statisztikai következtetések levonására alkalmazunk, ellenoriznünk kell, hogy a **lineáris regresszió alapvető előfeltételei** teljesülnek-e modellünkre.

Éppen ezért fontos, hogy elemzésünk minden fontosabb modelljét ellenorizzuk. Ez mindenképpen érinti a végső modellünket, de gyakran érdemes a modellválasztás során épített köztes modelleket is ellenorizni.

Fontos megjegyezni, hogy ha bármit változtatunk a modellünkön, vagy az adatainkon a modelldiagnosztika eredményei alapján, úgy a diagnosztikát **újra el kell végeznünk**.

3.1 Modell előállítás

Elso lépésként állítsunk elő egy modellt, amely pusztán az `sqm_living` és `grade` változók alapján megállapítja az ingatlan árát.

Futassuk ezen a modellen a modell diagnosztikát!

```
mod_house2 <- lm(price_mill_HUF ~ sqm_living + grade, data = data_house)
```

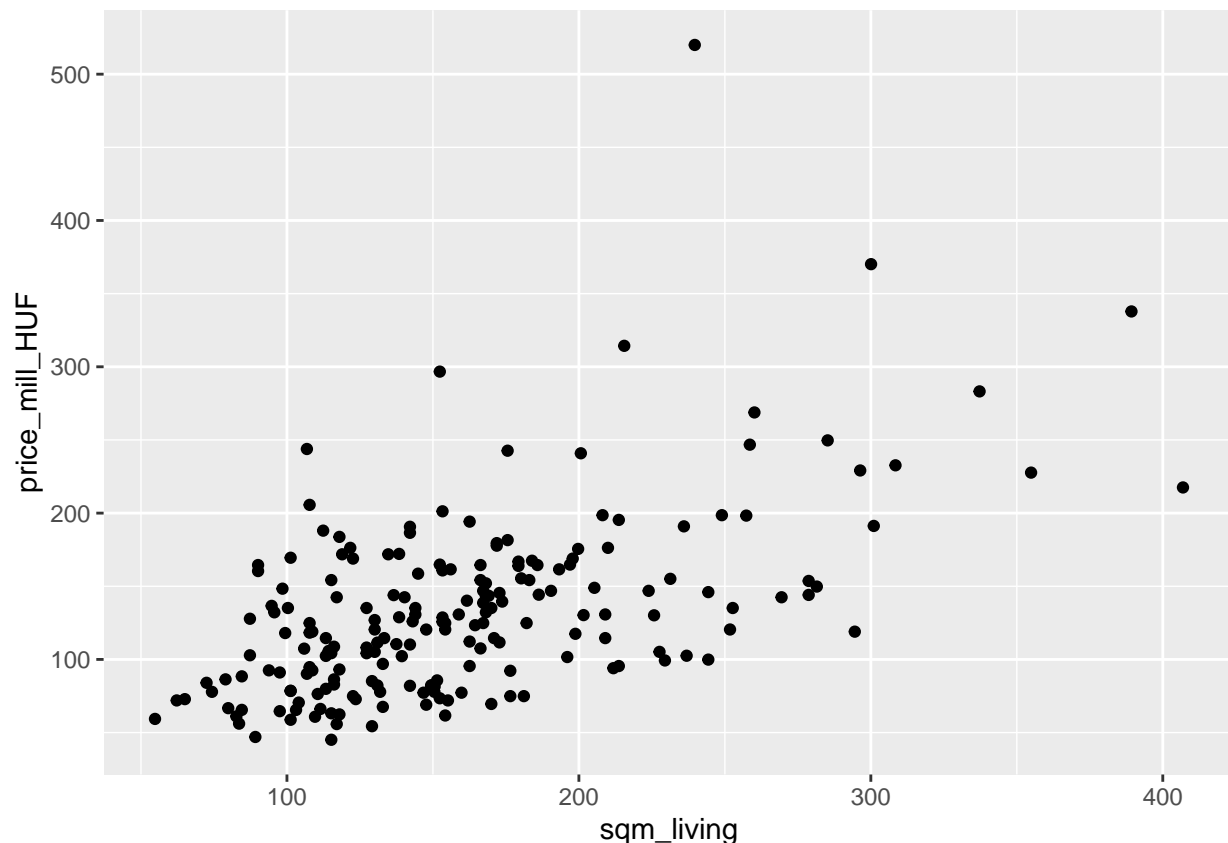
4 Kiugró adatok kezelése

4.1 Szélsőséges esetek azonosítása

A szélsőséges eseteket azonosíthatjuk a kimeneti változók azonosíthatjuk például az **adatok vizualizációja** során.

Példánkban az ár és alapterület adatait ábrázoljuk scatterplot-on.

```
data_house %>%  
  ggplot() + aes(x = sqm_living, y = price_mill_HUF) + geom_point()
```



Latható hogy a legtöbb ingatlan végső ára 200 millió forint alatt volt, de voltak kivételek is. Az 200 millió forintnál drágább ingatlanokat tekinthetjük szélsőséges értékeknek, különösképp az 500 millió forint árú ingatlant.

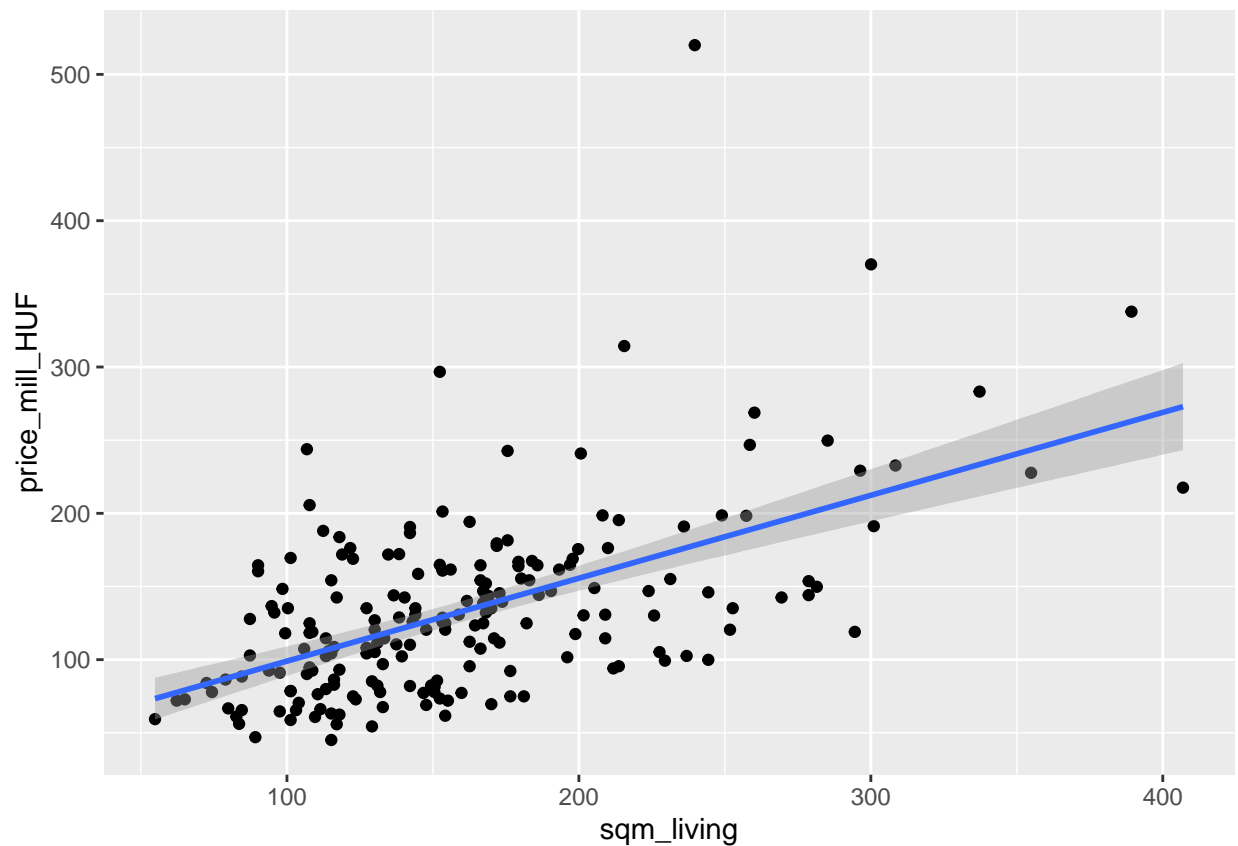
Nem szükséges azonban eltávolítani ezeket az adatokat ha van elég pontunk ami ellensúlyozhatja ezeknek a hatását.

4.2 Jelentős hatású kiugró értékek azonosítása

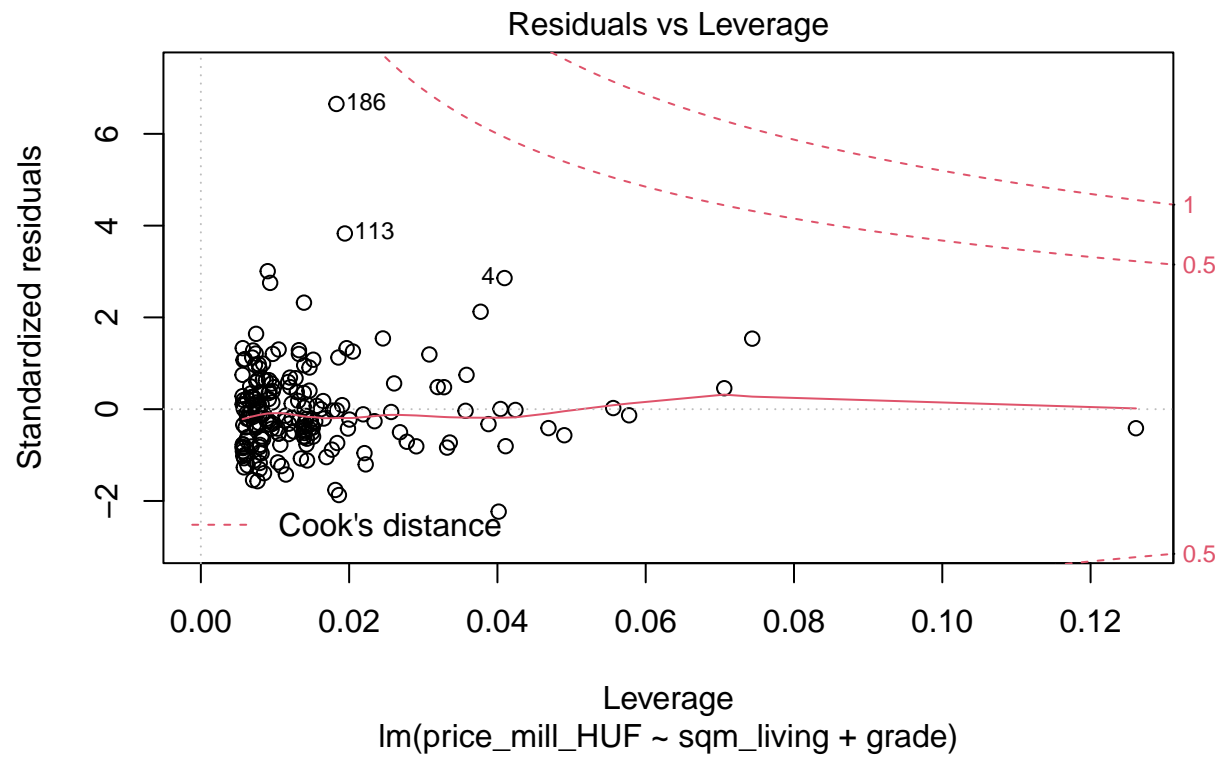
A helyzet bonyolultabb azokban az esetekben ha az érték nem csak jelentősen eltér a többi adattól, de **a regressziós vonalra is számottevő hatással bír**. Ezeket nagy befolyású eseteknek nevezzük (**high leverage cases**). Ezeket a nagy befolyású eseteket a scatter plot-ot vizsgálva, a residual-leverage plot segítségével, és a Cook távolságon keresztül fedezhetjük fel.

```
data_house %>%  
  ggplot() + aes(x = sqm_living, y = price_mill_HUF) + geom_point() +  
  geom_smooth(method = "lm")
```

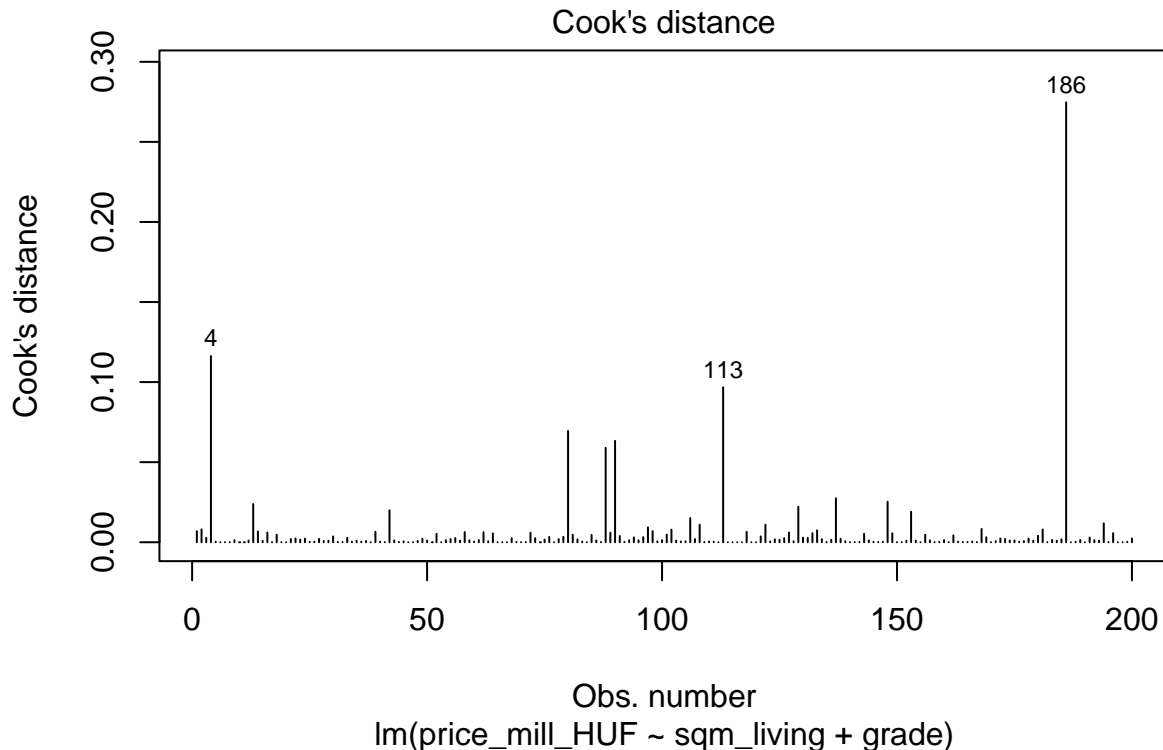
```
## `geom_smooth()` using formula 'y ~ x'
```



```
mod_house2 %>%  
  plot(which = 5)
```



```
mod_house2 %>%  
  plot(which = 4)
```



Azok a pontok, melyek a diagrammon a regressziós vonal közepéhez közel helyezkednek el kisebb befolyással vannak arra mint a **végeknél lévoek**. Azok az esetek (megfigyelesek) amelyek nagy reziduális hibával és nagy befolyással jellemezhetőek, nagy hatást fejtenek ki a modellre. A **Cook távolság** mutatja meg, mekkora egy eset hatása a modellre.

Bár nincs konkrét szabály a problémás esetek meghatározására, de van néhány általános alapelv. Vannak akik az **1-nél nagyobb** Cook távolságú értékeket, míg mások a **4/N-nél** (ahol N az adatok száma) **nagyobb** Cook távolságot tekintik jelentős hatásúnak.

Esetünkben egyetlen esetben sem nagyobb a Cook távolság 1-nél, néhány esetben azonban 0,02-t már meghaladja. Vagyis a második kirtérium alapján van néhány nagy hatású eset a mintában.

A nagy hatású esetek jelenlete onmagában nem feltétlenül jelent orvosolando problemat, viszont ez könnyen vezethet a regressziós modell **bejoslo erejenek csokkenesehez**, es ahhoz, hogy a regresszio alapfeltetelei megserulnek.

Eloszor is teszteljük hogy teljesülnek-e a többszörös regresszió előfeltetelei, és csak utána hozzunk döntést azzal kapcsolatban, hogy mit kezdünk a nagy hatású esetekkel.

5 A lineáris regresszió előfeltetelei

- **Normalitás:** A modell rezidualisai normáeloszlást követnek
- **Linearitás:** A prediktor és a kimeneti változó között lineáris kapcsolat kell legyen
- **Homoszkedaszticitás:** A rezidualisok varianciája minden értékre hasonló a prediktorokéhoz
- **Nincs kollinearitás:** egyetlen prediktor sem határozható meg a többi prediktor lineáris kombinációjaként.

5.1 Normalitás

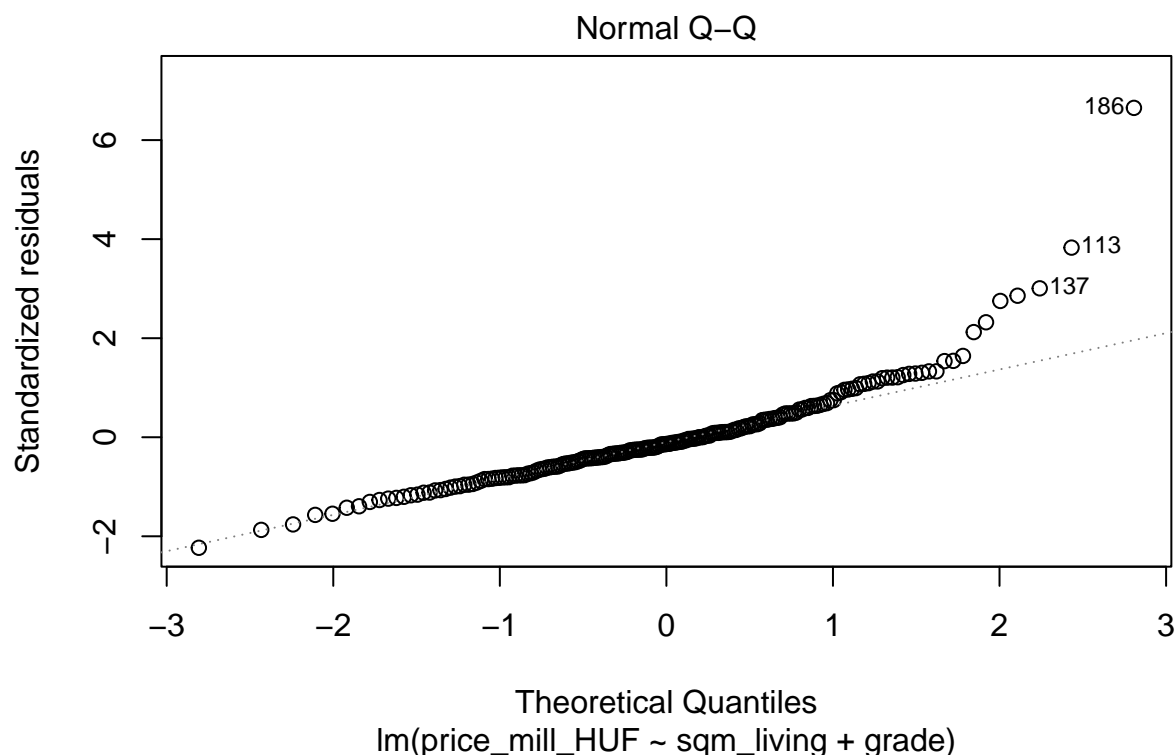
A modell **rezidualisai normáeloszlást** kell kövessenek. Megjegyzendo, hogy itt a modellbol származó előrejelzés hibájáról (rezidualisáról), és nem az egyes prediktorok vagy bejósolt változó eloszlásáról beszélünk.

Ezt a előfeltetelt egy QQ diagramm (**QQ plot**) ábrázolásával, és az esetek elméleti, diagonálshoz viszonyított elrendezésének vizsgálatával ellenorizhetjük. Ha az esetek jelentosen eltérnek az ábrán szaggatott vonallal jelzett elméleti diagonálistól, úgy a normalitásra vonatkozó előfeltétel sérülhet.

A rezidualisok **hisztogrammját** is érdemes szemügyre vennünk. Ezen egy a normál eloszlásnak megfelelo, Gauss-görbéhez hasonló alakzatot kell látnunk ha a normalitás feltétele teljesül.

A **skew és kurtosis** statisztikákat is lekérdezhethetjük a `describe()` függvény segítségével. ha a skew és kurtosis > 1 , úgy az a normalitási feltétel sérülését jelezheti.

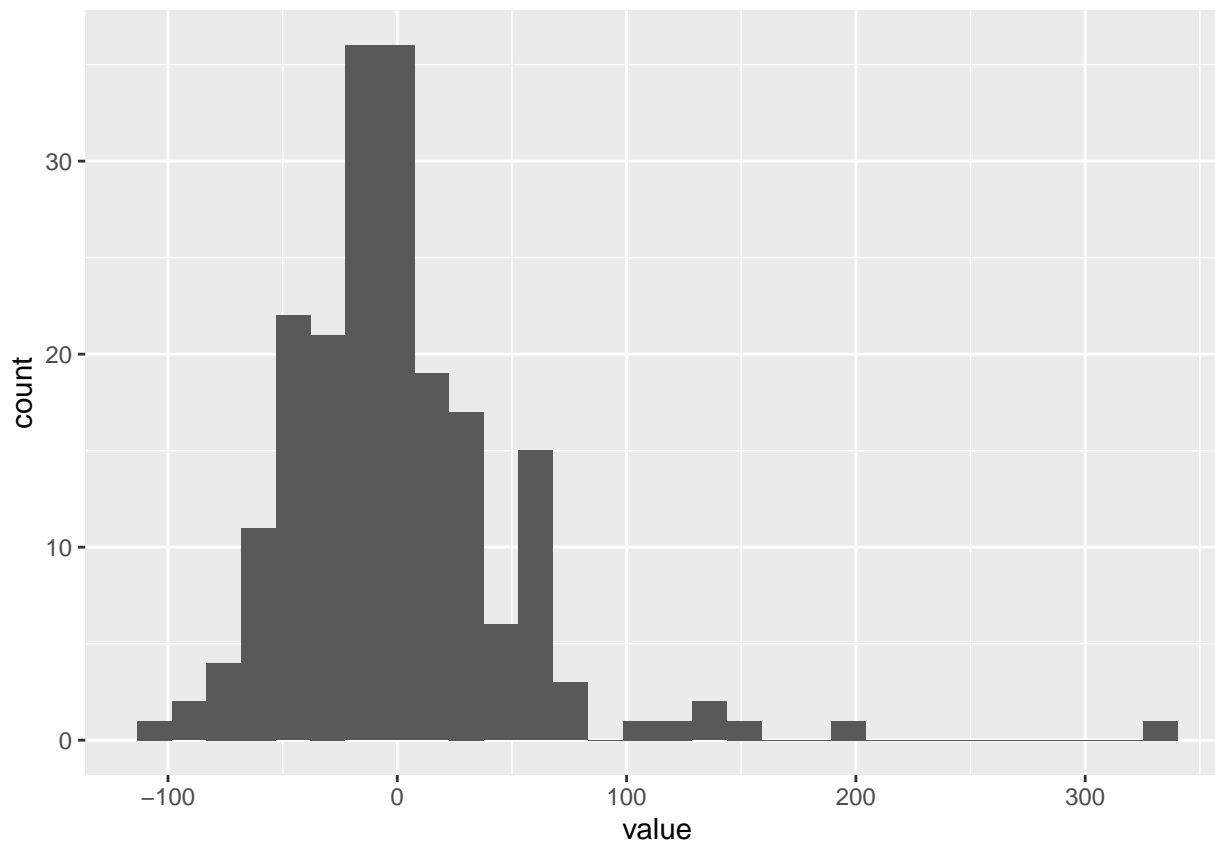
```
# QQ plot
mod_house2 %>%
  plot(which = 2)
```



```
# histogram

residuals_mod_house2 = enframe(residuals(mod_house2))
residuals_mod_house2 %>%
  ggplot() + aes(x = value) + geom_histogram()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# skew and kurtosis
describe(residuals(mod_house2))

##      vars   n mean    sd median trimmed  mad      min     max range skew kurtosis
## X1      1 200    0 49.71  -6.79  -4.31 36.98 -109.26 329.24 438.5 2.05      9.73
##      se
## X1 3.52
```

Az eredmények alapján látható hogy a rezidualisok enyhén eltérnének a normalitási feltételtől, ami elsősorban a néhány problémás esetnek tudható be.

5.1.1 Mi történik a normalitási feltétel sérülése esetén?

A becslések és **konfidencia intervallumok pontossága** a normalitási feltétel sérülése esetén csökkenhet. Ennek mértéke a **minta méretétől** is függ. Nagy minták esetén ($N > 500$) a hatás szinte elhanyagolható, míg kisebb minták esetén (N kb. 100) a hatás nagyobb. Lumley, Diehr, Emerson és Chen (2002) kutatásában például a normalitás szélsőséges sérülése esetén ($\text{skewness} = 8,8$; $\text{kurtosis} = 131$) a 95%-os konfidencia intervallum a szimulációk 93,6%-ában tartalmazta a populációatlag $N=500$ -as minta esetén, és 91,3%-ában az $N=65$ -ös esetben.

Következik tehát, hogy a normalitási feltétel sérülése esetén a konfidencia intervallumok és p értékek kevésbé lesznek megbízhatóak, de ennek figyelembevételével továbbra is felhasználhatóak.

Hivatkozások:

Lumley, T., Diehr, P., Emerson, S., & Chen, L. (2002). The importance of the normality assumption in large public health data sets. *Annual review of public health*, 23(1), 151-169.

5.1.2 Mit tegyünk, ha a normalitási feltétel sérül?

1. Kezelhetjük egyszerűen óvatosabban eredményeinket, például **99%-os konfidencia intervallum** használatával a szokásos 95%-os helyett, vagy tekinthetjük $p < 0,01$ -et a szignifikancia határának.
2. Megpróbálhatjuk a prediktorainkat vagy a bejósolt változót úgy **transzformálni**, hogy rezidualisaink eloszlása közelebb legyen a normál eloszláshoz. Ekkor azonban fontos figyelembe venni, hogy az így kapott együtthatók is transzformálva lesznek. Ugyan ez vonatkozik a hiba feltételekre, azaz ha a modell transzformált értékekre vonatkozó RSS-je nem lesz összehasonlítható az transzformatlan értékekével. Az átalakításról további információk az alábbi linken találhatóak: http://abacus.bates.edu/~ganderso/biology/bio270/homework_files/Data_Transformation.pdf (a file szerzője számomra ismeretlen, de a dokumentum tartalmilag pontos, és megfelelő hivatkozásokkal ellátott).
3. Ha mindössze néhány eset okozza a normalitástól való eltérést, úgy hasznos lehet a **kiugró értékek kizárása**. Formális hipotézisteszt esetén a változók kizárása nem alapulhat a p-értéken. A kizárás feltételei preregisztrálhatóak, vagy egy érzékenységi elemzést is használhatunk, azaz az adott elemzést kétszer lefuttathatjuk adatainkon, egyszer a problémás értékek bevonásával, egyszer pedig azok kizárásával, majd az eredményeket összehasonlíthatjuk a két esetben.

Esetünkben, lévén adataink mindössze néhány kiugró eset miatt sértik a normalitási feltételt, megpróbálhatjuk kizárni ezeket az adatokat, hátha így kiküszöbölhető a probléma. Itt a **186-os és 113-as esetek** kizárását választottuk azok Cook távolsága alapján, és mivel a **QQ plot** alapján szerepük volt a normalitástól való eltérésben.

Az alábbiakban a fenti két eset kizárásával újra illesztjük modellünket, és újra ellenőrizzük a normalitási feltételt. Látható, hogy a kiugró adatok nélkül a rezidualisok a normál eloszláshoz lényegesen hasonlóbb eloszlást mutatnak mint korábban.

```
data_house_nooutliers = data_house %>%
  slice(-c(186, 113))

mod_house3 = lm(price_mill_HUF ~ sqm_living + grade, data = data_house_nooutliers)

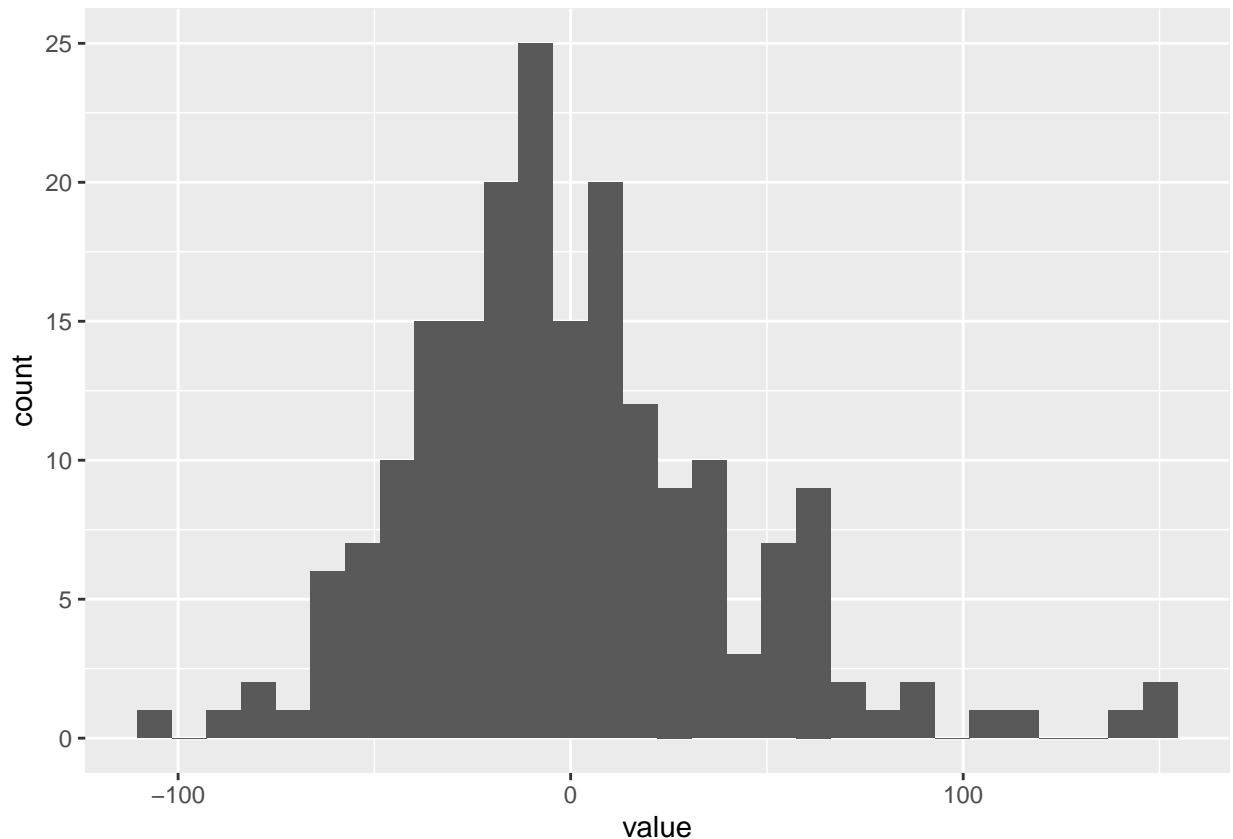
# recheck the assumption of normality of residuals
describe(residuals(mod_house3))

##      vars      n mean      sd median trimmed   mad      min max   range skew kurtosis
## X1         1 198    0 41.87  -4.99   -2.52 35.66 -102.13 154 256.13 0.79      1.35
##          se
## X1 2.98

residuals_mod_house3 = enframe(residuals(mod_house3))

residuals_mod_house3 %>%
  ggplot() + aes(x = value) + geom_histogram()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Amikor a két modellt összehasonlítjuk, látható hogy a kiugró adatok kizárása nem változtatott a statisztikai következtetéseinket, hisz a korábban jelentős prediktorok továbbra is jelentősek, és a modell F próbája is szignifikás mindkét esetben. Az **adjusted R² érték lényegesen javult**, hisz a regressziós vonalunk mostmár sokkal jobban illeszkedik a megmaradó adatainkra.

(Ugyanakkor a modell illeszkedését új adatokon, vagy egy **test-seten** is érdemes kipróbálnunk, hogy az előrejelzéseink hatékonyságáról tisztább képet alkothassunk, hiszen a kizártakhoz hasonló kiugró értékek az új adatok között is szerepelhetnek, melyek meghatározásában modellünk pontatlan lesz.)

```
# comparing the models on data with and without the
# outliers
summary(mod_house2)
```

```
##
## Call:
## lm(formula = price_mill_HUF ~ sqm_living + grade, data = data_house)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -109.26  -29.55   -6.79   19.65  329.24
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -51.2305    27.9831  -1.831  0.068646 .
## sqm_living     0.3768     0.0783   4.813  2.96e-06 ***
## grade         16.8485     4.7158   3.573  0.000444 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.96 on 197 degrees of freedom
## Multiple R-squared:  0.358, Adjusted R-squared:  0.3515
## F-statistic: 54.94 on 2 and 197 DF,  p-value: < 2.2e-16

summary(mod_house3)

##
## Call:
## lm(formula = price_mill_HUF ~ sqm_living + grade, data = data_house_nooutliers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.130  -28.175   -4.994   21.582  154.004
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -48.21536   23.84403  -2.022  0.0445 *
## sqm_living    0.34451    0.06614   5.208 4.82e-07 ***
## grade        16.78639    4.01091   4.185 4.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.08 on 195 degrees of freedom
## Multiple R-squared:  0.413, Adjusted R-squared:  0.407
## F-statistic: 68.59 on 2 and 195 DF,  p-value: < 2.2e-16
```

A további feltétel-vizsgálatokban azt a modellt vizsgáljuk majd, amiből már kizartuk ezeket a kiurgo eseteket (186-os és 113-as esetek).

4. **Bootstrapping** módszert is használhatjuk a konfidencia szintek robosztus becslésére a normalitás feltétel sérülése esetén.

5.1.3 Bootstrapping

A bootstrapping lenyege hogy **saját mintánkból véletlenszerűen mintákat veszünk**, és ezeken az új mintakon illesztjük újra a modellünket. Ezt a folyamatot **sokszor megismételjük** (1000-10000 alkalommal), majd ezek eredményei alapján következtetünk a konfidencia határookra.

(Az alábbiak megfelelő működéséhez szükséges a fenti saját függvények futtatása.)

Hasonlítsuk össze a szokásos és a bootstrapping módszerrel nyert konfidencia intervallumokat.

```
# regular confidence intervals for the model coefficients
confint(mod_house3)
```

```
##              2.5 %      97.5 %
## (Intercept) -95.240646 -1.1900718
## sqm_living    0.214059  0.4749585
## grade        8.876049 24.6967253
```

```
# bootstrapped confidence intervals for the model coefficients
confint.boot(mod_house3)
```

```
##              boot 2.5 % boot 97.5 %
## (Intercept) -97.0148873 -0.6843126
## sqm_living    0.2115049  0.4863262
```

```
## grade          9.1199482  24.0827281
# regular adjusted R squared
summary(mod_house3)$adj.r.squared

## [1] 0.406965
# bootstrapping with 1000 replications
results.boot <- boot(data = data_house, statistic = adjR2_to_boot,
  R = 1000, model = mod_house3)

# get 95% confidence intervals for the adjusted R^2
boot.ci(results.boot, type = "bca", index = 1)

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results.boot, type = "bca", index = 1)
##
## Intervals :
## Level      BCa
## 95%      ( 0.2284,  0.4790 )
## Calculations and Intervals on Original Scale
```

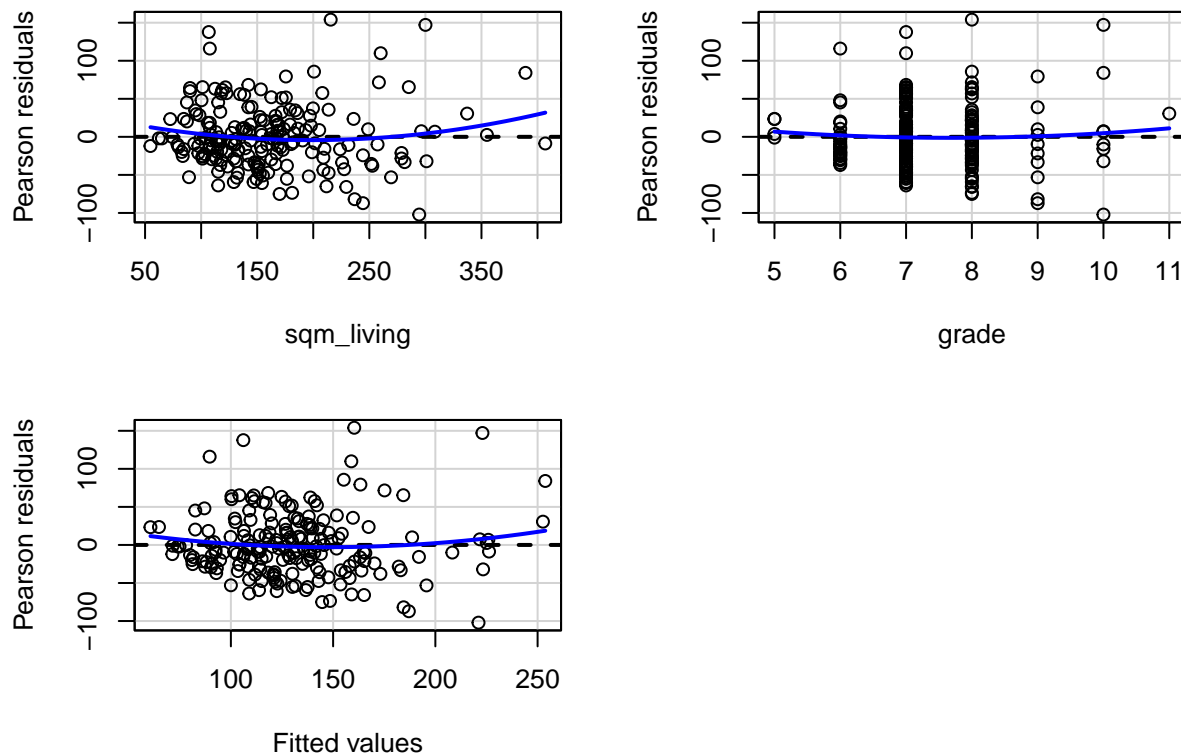
5.2 Linearitás

Az eredmény és a prediktorok között lineáris kapcsolat kell legyen.

A car csomag `residualPlots()` függvényének prediktoronkénti használatával vizsgálhatjuk meg a linearitást. A függvény eredményeként egy **scatterplot-ot kapunk egy spline-al**, mely dűrván jelzi a prediktor és az eredmény közti kapcsolatot, illetve a rezidualis-elorejelzett érték plot-ot is megkapjuk. A linearitás teljesülése esetén az összes kapott ábrán megközelítőleg vízszintes vonalakat kell látnunk.

A **`residualPlots()` függvény** a linearitás serulesenek tesztjét is elvégzi. A teszt szignifikanciája esetén ($p < 0.05$) arra következtethetünk, hogy a linearitási feltétel sérül.

```
mod_house3 %>%
  residualPlots()
```



```
##           Test stat Pr(>|Test stat|)
## sqm_living    1.6218      0.1065
## grade         0.6485      0.5174
## Tukey test    1.2576      0.2085
```

Esetünkben, bár látható némi görbület az árakon, a tesztek egyike sem szignifikáns, vagyis a modellünk feltehetően eleget tesz a linearitási elvárásnak.

5.2.1 Mi a hatása a linearitás sérülésének?

Ha a prediktorok és bejósolt változók között nem lineáris a kapcsolat, úgy **modellünk előrejelzései pontatlanok** lehetnek. Továbbá a modell együtthatói is megbízhatatlanok lesznek ha előrejelzéshez szeretnénk használni őket. Például a linearitás feltételének sérülése esetén előfordulhat hogy a standardizált együtthatók, t-próbák és p-értékek azt sejtethetik, hogy egy prediktornak nincs hatása a kimenetre, lehet hogy valójában a prediktor mégis hordoz releváns információt a bejósoláshoz, csak az összefüggés nem lineáris.

5.2.2 Mit tegyünk ha a linearitás sérül?

A linearitás sérülése esetén modellünk rugalmasabbá tételével érdemes próbálkoznunk.

1. egyik lehetőség a **hatványprediktorok** használata. (Ennek pontos menetét a speciális prediktorok gyakorlatban tárgyaltuk.) Általában a másod és harmadrendű hatványprediktorok használata már eléggé bizonyul. Érdemes elkerülni hogy túl magasrendű hatványtényezőt tegyünk a modellünkbe, ugyanis az overfitting-hez vezethet.
2. ha a hatványprediktorok nem alkalmasak az összefüggés leírására, érdemes lehet a **nem lineáris regresszióval** próbát tenni. Ez nem része a tananyagnak ezen a szinten. Akit érdekel, az az alábbi könyvben olvashat erről a módszerrel:

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Ingyen hozzáférhető itt: <http://www-bcf.usc.edu/~garth/ISL/>

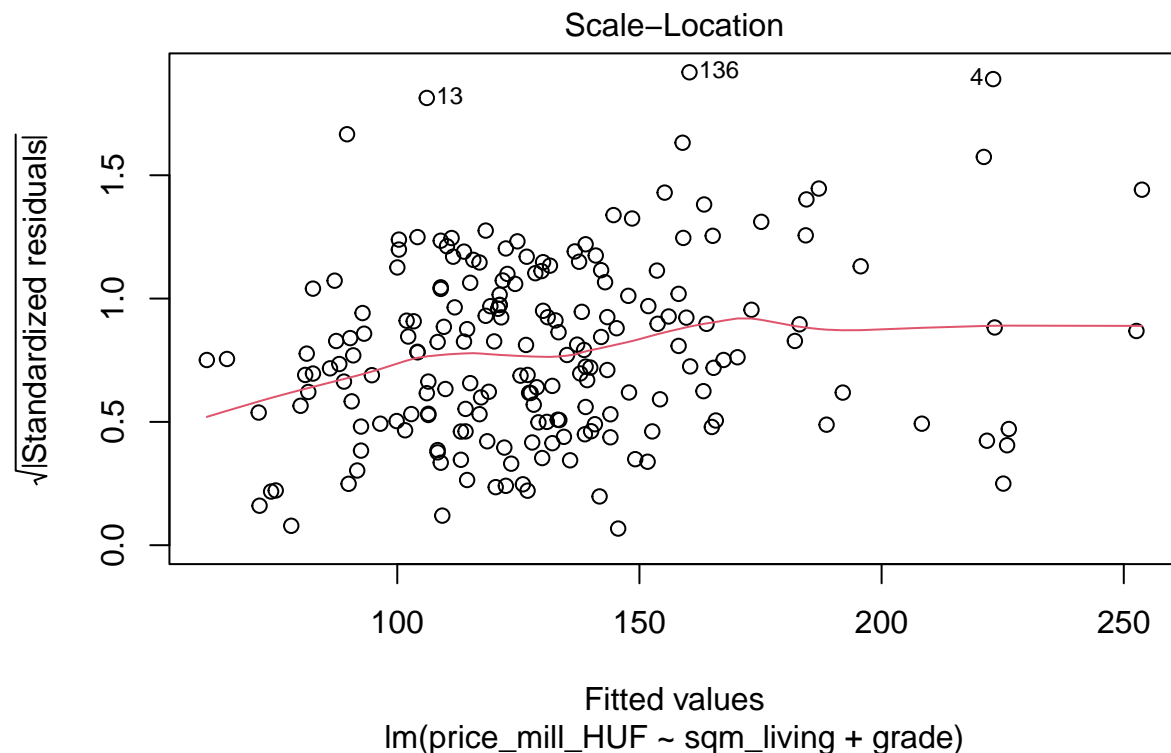
5.3 Homoszkedaszticitás

A regresszió során feltételezzük, hogy a **hiba tagok (rezidualisok) szórása konstans és a prediktorok értékétől független**. Tehát például a rezidualisok varianciájának 60 négyzetmeter alapterületű és 300 négyzetmeter alapterületű ingatlanok esetén meg kell egyezzen.

Ezt a standardizált **rezidualisokat és a prediktorokat ábrázoló diagram** vizsgálatával ellenőrizhetjük, ahol azt várjuk, hogy megközelítőleg azonos varianciát figyelünk majd meg minden előrejelzett érték esetén. Hasznos statisztikai próbák is rendelkezésünkre állnak. Így például a **Breush-Pagan tesztet** a `bptest()` függvénnyel hívhatjuk meg, melyet a `lmtest` csomagban találhatunk. Egy másik lehetőség az **NCV teszt** amely az `ncvTest()` függvénnyel hívható meg, és az R-nek eleve részét képezi (nincs szükség további csomagra). Ha a p -érték < 0.05 ezekben a tesztekben, úgy a homoszkedaszticitás sérülésére (vagyis jelentős heteroszkedaszticitásra) következtethetünk.

A fent említett tesztek alapján jelentős heteroszkedaszticitást találunk a `mod_house3` modellben, így ezt további vizsgálatnak kell alávetnünk.

```
mod_house3 %>%  
  plot(which = 3)
```



```
mod_house3 %>%  
  ncvTest() # NCV test
```

```
## Non-constant Variance Score Test
```



```
## Variance formula: ~ fitted.values
## Chisquare = 17.01078, Df = 1, p = 3.7168e-05

mod_house3 %>%
  bptest() # Breush-Pagan test

##
## studentized Breusch-Pagan test
##
## data: .
## BP = 10.028, df = 2, p-value = 0.006645
```

5.3.1 Mi a helyzet a heteroszkedaszticitás esetén?

Amennyiben heteroszkedaszticitás lép fel, úgy **modellünk pontatlan lehet**. Ettől függetlenül **használható marad** a modell, egyszerűen csak pontatlanabbul határozhatjuk meg az új adatainkat.

Ami fontosabb, hogy a modell együtthatói és a hozzájuk tartozó **konfidencia intervallumok is pontatlanok** lesznek.

Ha tehát szeretnénk meghatározni az ingatlanok értékeit a prediktorok alapján, azt bár pontatlanabbul, de a heteroszkedaszticitás fennállása mellett is megtehetjük, de az egyes együtthatók és a konfidencia intervallumok megbízhatósága serül.

5.3.2 Mit tehetünk a heteroszkedaszticitás orvoslására?

Ebben az esetben az alábbi módszerek bizonyulhatnak célravezetőnek:

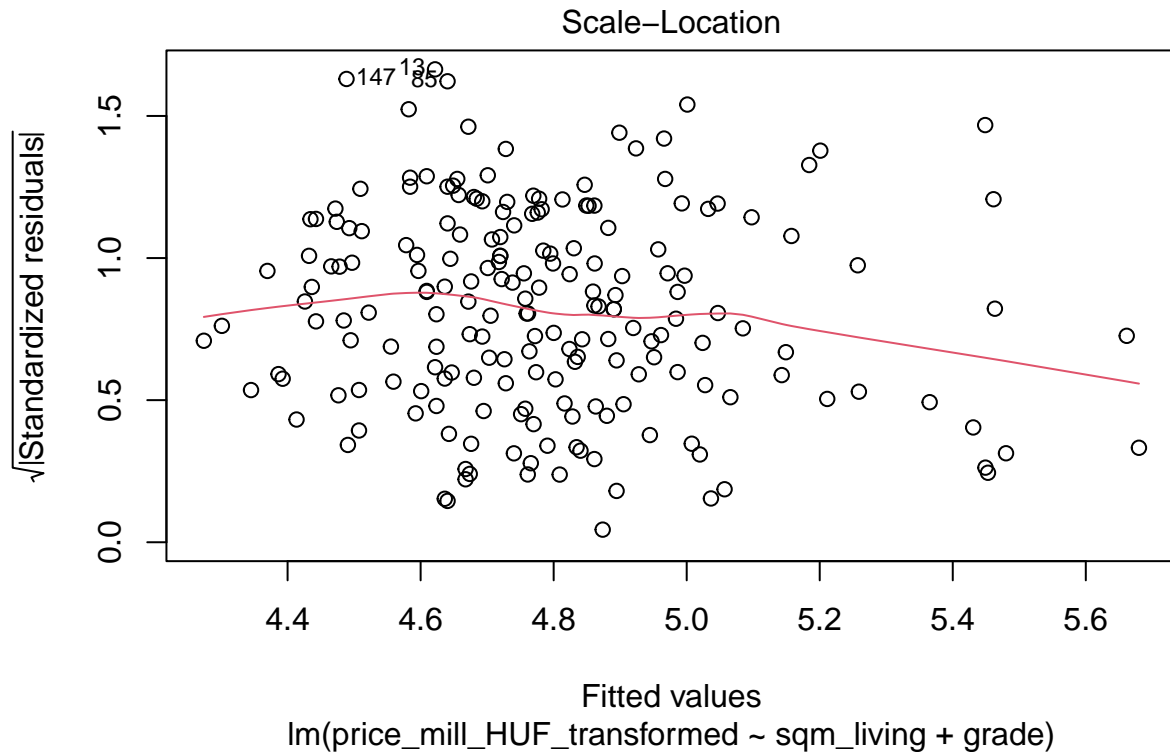
1. **Transzformáció.** ha elsődleges célunk az, hogy javítsuk előrejelzéseink pontosságát, úgy ezt a **kimeneti értékek és/vagy prediktorok** transzformációjával elérhetjük, azok **normál eloszlásúvá tételével**, így homogenizálva a varianciát az adatsor bizonyos részeire. Például alább a **log()** transzformációt alkalmazzuk a bejosolt változóra, vagyis az ingatlanok árára, majd ennek megfelelően újra illesztjük modellt. Ennek az eljárásnak az eredményeként a heteroszkedaszticitás tesztek már nem szignifikánsak.

Ne feledkezzük meg azonban arról, hogy mostmár nem az árakat, hanem azok logaritmusát határozzuk meg, ezért a kapott eredményeket vissza kell még transzformálni az exponenciális “exp()” függvénnyel. A modell együtthatók meghatározásánál is fontos, hogy az ár értékek helyett azok logaritmusait használtuk.

```
data_house_nooutliers = data_house_nooutliers %>%
  mutate(price_mill_HUF_transformed = log(price_mill_HUF))

mod_house4 = lm(price_mill_HUF_transformed ~ sqm_living + grade,
  data = data_house_nooutliers)

mod_house4 %>%
  plot(which = 3)
```



```
mod_house4 %>%
  ncvTest() # NCV test
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.5274885, Df = 1, p = 0.46766
```

```
mod_house4 %>%
  bptest() # Breush-Pagan test
```

```
##
## studentized Breusch-Pagan test
##
## data: .
## BP = 0.97249, df = 2, p-value = 0.6149
```

A kapott eredményeket az `exp()` függvénnyel transzformálhatjuk vissza az eredeti skálára.

```
exp(predict(mod_house4))
```

2. **Robosztus becslés alkalmazása.** ha fontos hogy megtartsuk az eredeti skálát hogy a modell egyutthatók megtarthassák intuitív jelentésük, használhatunk robusztus közelítési módszereket a heteroszkedaszticitás-konzisztens (HC) standard hibák meghatározására, és használhatjuk ezeket a konfidencia intervallumok, és a modell egyutthatóira vonatkozó korrigált p-értékek meghatározására. Az így kapott értékek kevésbé érzékenyek a heteroszkedaszticitásra, ezért nevezzük őket robusztus becsléseknek. Az alábbi példában a **Huber-White Sandwich becslést** használjuk.

Kis minták esetén (N kb. 50) a standardizált hiba korrigálásához más módszerre lehet szükség, például a **Bell-McCaffrey féle közelítésre**. Ennek részletesebb leírását lásd Imbens és Kolesar (2016) cikkében,

az általuk használt R függvényeket pedig az alábbi linken: <https://github.com/kolesarm/Robust-Small-Sample-Standard-Errors>

Hivatkozások:

Imbens, G. W., & Kolesar, M. (2016). Robust standard errors in small samples: Some practical advice. *Review of Economics and Statistics*, 98(4), 701-712.

```
# compute robust SE and p-values
mod_house3_sandwich_test = coeftest(mod_house3, vcov = vcovHC,
  type = "HC")
mod_house3_sandwich_test

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -48.215359  24.025852 -2.0068   0.04615 *
## sqm_living   0.344509   0.067113  5.1332 6.864e-07 ***
## grade        16.786387   3.819320  4.3951 1.817e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

mod_house3_sandwich_se = unclass(mod_house3_sandwich_test)[,
  2]

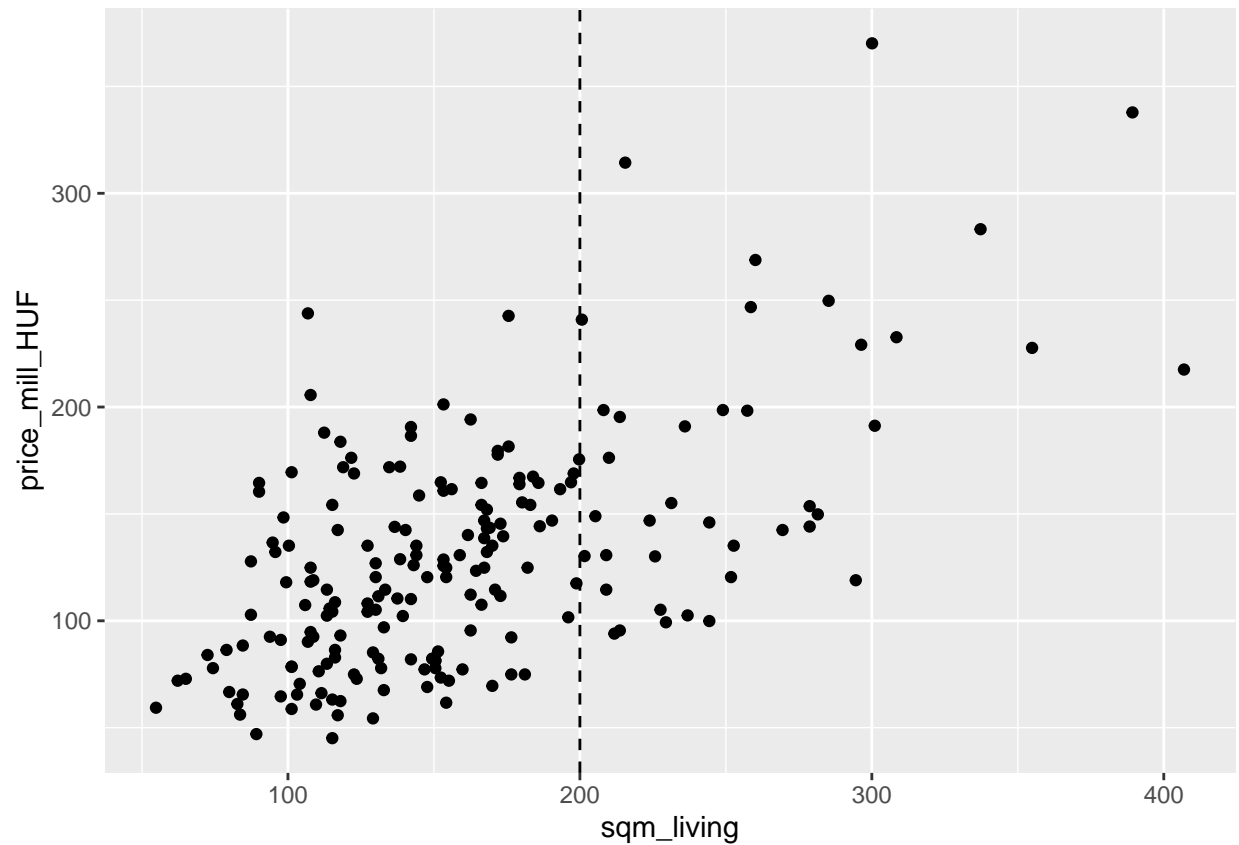
# compute robust confidence intervals
CI95_lb_robust = coef(mod_house3) - 1.96 * mod_house3_sandwich_se
CI95_ub_robust = coef(mod_house3) + 1.96 * mod_house3_sandwich_se

cbind(mod_house3_sandwich_test, CI95_lb_robust, CI95_ub_robust)

##              Estimate Std. Error  t value      Pr(>|t|) CI95_lb_robust
## (Intercept) -48.2153590 24.02585161 -2.006812 4.615088e-02   -95.3060282
## sqm_living   0.3445087  0.06711329  5.133242 6.864072e-07    0.2129667
## grade        16.7863869  3.81931981  4.395125 1.816739e-05    9.3005201
##              CI95_ub_robust
## (Intercept)   -1.1246899
## sqm_living     0.4760508
## grade         24.2722538
```

3. **Külön modellek az adatsor eltérő részeihez.** Egy újabb módszert is bevethetünk, ha a variancia valamilyen egyszerű mintázat / csoportosulás szerint mutat heteroszkedaszticitást. A mi mintánk esetében megfigyelhető, hogy a 200 négyzetméteres alapterületű ingatlanok esetén hirtelen megnő a variancia. Az adatainkat tehát kettéválasztjuk 200 négyzetméteres és az alatti alapterületű (data_house_small), illetve annál nagyobb alapterületű ingatlanokra (data_house_large), majd ezekre külön-külön modelleket illesztünk. Így ha egy 200 négyzetméter, vagy az alatti lakás értékét szeretnénk meghatározni, úgy a mod_house3_small, míg ha ennél nagyobb ingatlanét, úgy a mod_house3_large modellt kell használnunk. A heteroszkedaszticitás tesztjei nem szignifikánsak ebben a két modellben.

```
data_house_nooutliers %>%
  ggplot() + aes(x = sqm_living, y = price_mill_HUF) + geom_point() +
  geom_vline(xintercept = 200, lty = "dashed")
```

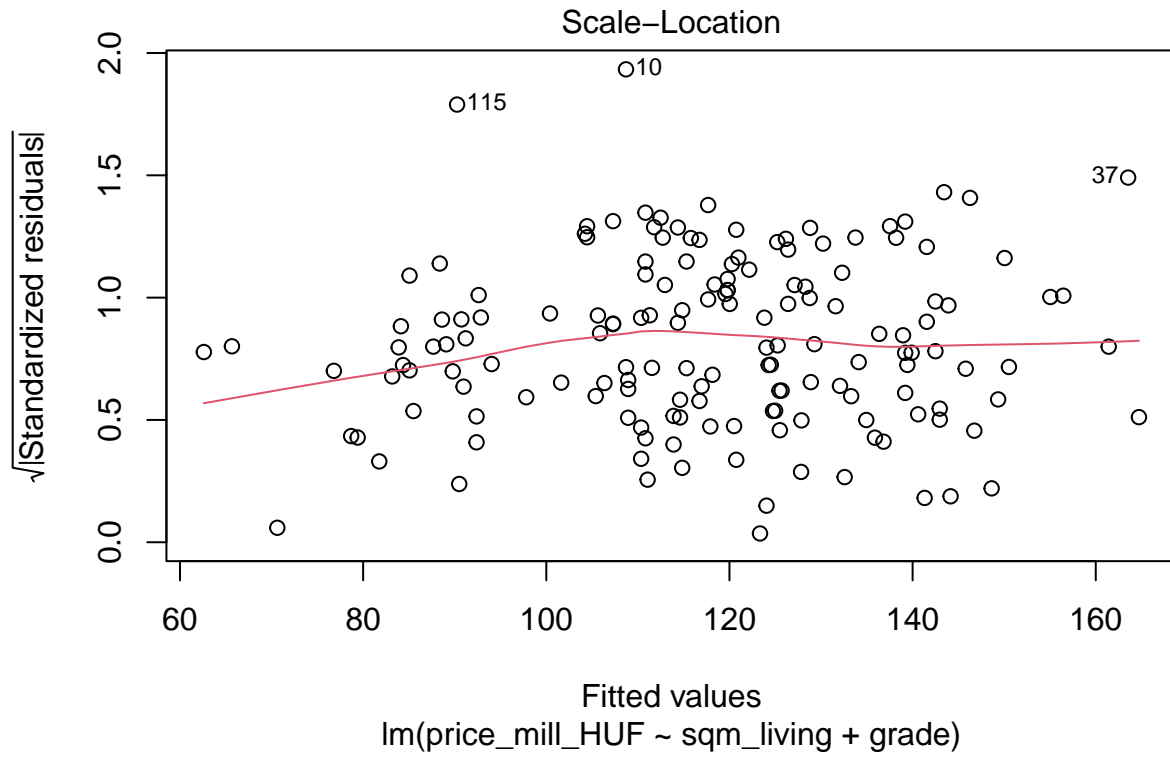


```
data_house_small = data_house_nooutliers %>%
  filter(sqm_living <= 200)

data_house_large = data_house_nooutliers %>%
  filter(sqm_living > 200)

mod_house3_small = lm(price_mill_HUF ~ sqm_living + grade, data = data_house_small)
mod_house3_large = lm(price_mill_HUF ~ sqm_living + grade, data = data_house_large)

mod_house3_small %>%
  plot(which = 3)
```



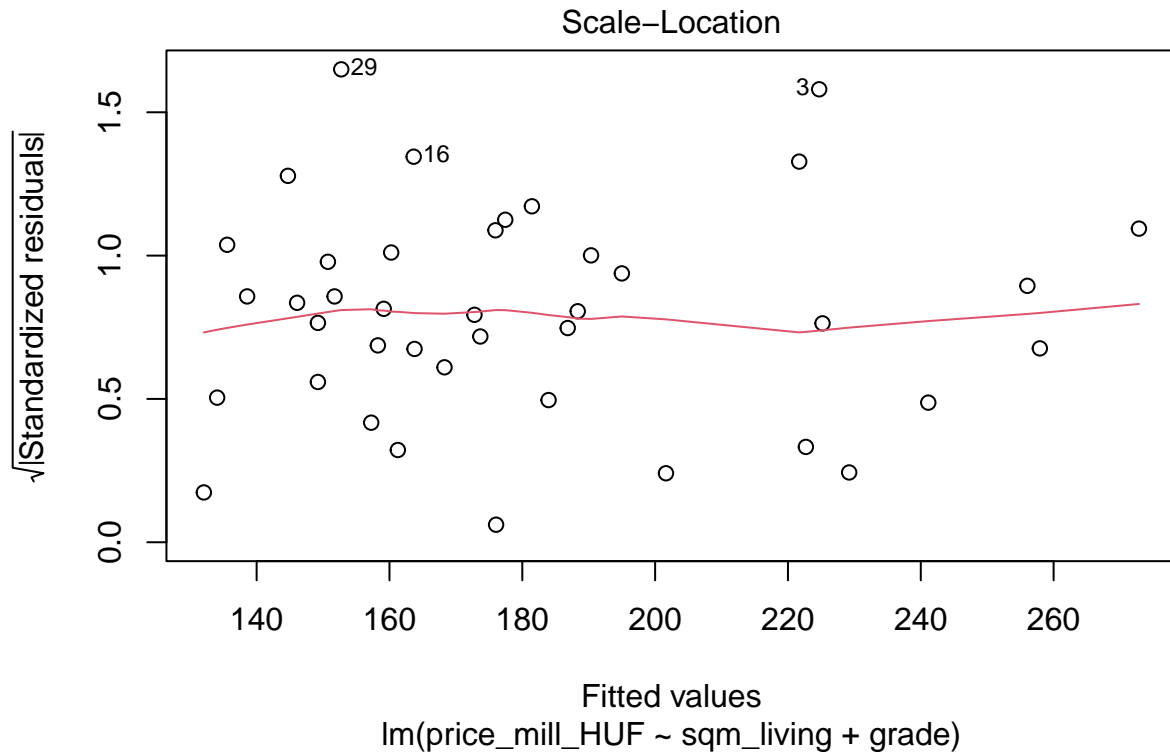
```
mod_house3_small %>%
  ncvTest() # NCV test
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.2195388, Df = 1, p = 0.63939
```

```
mod_house3_small %>%
  bptest() # Breush-Pagan test
```

```
##
## studentized Breusch-Pagan test
##
## data: .
## BP = 0.95326, df = 2, p-value = 0.6209
```

```
mod_house3_large %>%
  plot(which = 3)
```



```
mod_house3_large %>%
  ncvTest() # NCV test
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.005016941, Df = 1, p = 0.94353
```

```
mod_house3_large %>%
  bptest() # Breush-Pagan test
```

```
##
## studentized Breusch-Pagan test
##
## data: .
## BP = 0.35366, df = 2, p-value = 0.8379
```

Ha egyszerre sérül a normalitás és a homoszkedaszticitás feltétele, akkor a **wild bootstrapping** ajánlott a sima bootstrapping helyett a konfidencia intervallum kiszámításához. (Ennek a funkciónak a futtatásához először le kell futtatni a script elején lévő kódot amivel elmentjük ezt a wild.boot.confint() funkciót.)

```
### get 95% confidence intervals with wild bottstrapping
wild.boot.confint(mod_house3)
```

```
##                2.5%        97.5%
## (Intercept) -95.6706601 -0.5856868
## sqm_living   0.2192235  0.4735917
## grade        9.2222052 24.0017299
```

5.4 A Multikollinearitás tesztelese

Feltételezzük, hogy a **prediktoraink lineárisan függetlenek** (egyik sem jelentősen bejósolható a többi prediktor ismerete alapján). Ez általában azt feltételezi, hogy az egyes prediktorok között **nincs erős korreláció**, ez azonban magában még kevés, hisz ha a prediktorok párosával esetleg még nem is mutatnak komoly korrelációt, ez magában még nem zárja ki azt hogy több prediktor bonyolult lineáris kombinációjaként előállhassanak.

A feltételezés ellenőrzése érdekében kiszámoljuk a variancia infláció faktort (VIF) minden prediktorra, a `vif()` függvény használatával, melyet a car csomagban találhatunk meg.

Arról, hogy mely **VIF értékek** jelölnek a kollinearitás szempontjából problémát, még nem született konszenzus. Vannak akik a 10 vagy afölötti vif értékeket tekintik problémásnak (pl.: Montgomery és Peck, 1992). Egy konzervatívabb megközelítés, ha a 3 feletti VIF értékek esetében már külön eljárunk. (Zuur, Ieno, és Elphick, 2010 javaslata alapján). A ZHk-ban használjuk a **3 feletti VIF értékeket**, mint kritériumot a multikollinearitás azonosítására.

Hivatkozások: Montgomery, D.C. & Peck, E.A. (1992) Introduction to Linear Regression Analysis. Wiley, New York. Zuur, A. F., Ieno, E. N., & Elphick, C. S. (2010). A protocol for data exploration to avoid common statistical problems. Methods in ecology and evolution, 1(1), 3-14.

```
mod_house3 %>%  
  vif()
```

```
## sqm_living      grade  
## 1.849221      1.849221
```

A fenti példában nincs probléma a kollinearitást illetően.

5.4.1 Mi a helyzet kollinearitás esetén?

Regresszió esetén gyakran szeretnénk tudni az egyes **prediktorok egyedi hozzáadott értéket** a modellhez, és hogy ez a hozzáadott érték statisztikailag **szignifikáns-e**. Az egyes prediktorokhoz tartozó regressziós együtthatók jelzik, a prediktor hatásának irányát és mértékét a többi prediktor hatásának fixen tartása mellett. Eroesen korreláló prediktorok esetén ritkán fog előfordulni, hogy az egyik magas, míg a többi alacsony, így az egyedi hatások is nehezen lesznek szétválaszthatóak.

Röviden összegezve tehát kollinearitás esetén a **modell együtthatói** és az egyedi predikciós értékeikre vonatkozó t-próbák **kevésbé lesznek megbízhatóak**. Továbbá a modell együtthatói kifejezetten instabilak lesznek, azaz a modell változásai esetén (pl.: egy prediktor eltávolítása esetén) nagyokat változhatnak, sőt akár **elojelet is válthatnak**.

Szerencsére az **elorejelzések pontosságát nem befolyásolja** a kollinearitás, így ha csak ez érdekel bennünket, nem is kell vele foglalkoznunk, csak ha érdekelnek minket az egyes regressziós együtthatók, és szeretnénk következtetni az egyes prediktorok egyedi hatásaira vagy modellhez való hozzájárulására.

5.4.2 Mi a teendő kollinearitás esetén?

A kollinearitásnak két formája van:

Szerkezeti kollinearitás, ebben az esetben egy olyan prediktort is hozzáadunk a modellünkhöz, mely egy vagy több másik prediktorból származik. Például hatvanyprediktorok (például `grade` és `grade^2`), vagy iterációk (pl.: `long*lat`).

Adat kollinearitás, ebben az esetben a kollinearitás magában az adatban jelenik meg, és nem csak a modellünk terméke.

Ezek vizsgálatához most két új modellt fogunk előállítani.

5.4.2.1 Szerkezeti kollinearitás kezelése Eloszor is állítsunk elo egy modellt, melyben az sqm_living és grade változokon túl a GPS koordinatakat is bevonjuk a modellunkbe mint prediktorokat.

Az elso modellben csak a prediktorok elsodleges hatásával fogunk foglalkozni, azaz nem lesznek interakciós elemek. A modell summary azt mutatja, hogy a hosszúság (long) egyutthatoja negatív, vagyis minél keletebbre megyünk, annál olcsóbbak lesznek az ingatlanok, ami logikus, hiszen a vizsgált terület nyugati részén helyezkedik el Seattle és az ocean. A szélességhez (lat) tartozó koeficiens pozitív, vagyis északra nő az ingatlanok ára. A szélesség prediktív értéke szignifikáns modellünkben. A VIF alapján nincs probléma multikollinearitással ebben a modellben.

```
mod_house_geolocation = lm(price_mill_HUF ~ sqm_living + grade +
  long + lat, data = data_house_nooutliers)
summary(mod_house_geolocation)
```

```
##
## Call:
## lm(formula = price_mill_HUF ~ sqm_living + grade + long + lat,
##     data = data_house_nooutliers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -70.384 -23.279  -4.365  16.446 142.030
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.022e+04  1.960e+03  -5.214 4.74e-07 ***
## sqm_living   3.794e-01  5.638e-02   6.730 1.89e-10 ***
## grade        1.547e+01  3.397e+00   4.555 9.29e-06 ***
## long        -2.208e+01  1.546e+01  -1.429   0.155
## lat          1.572e+02  1.838e+01   8.551 3.72e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.53 on 193 degrees of freedom
## Multiple R-squared:  0.5858, Adjusted R-squared:  0.5772
## F-statistic: 68.25 on 4 and 193 DF,  p-value: < 2.2e-16

mod_house_geolocation %>%
  vif()

## sqm_living      grade      long      lat
##  1.884575  1.860764  1.041486  1.024952
```

Most helyezzük el az interakciós tagot is modellünkben (a * jelet használva a + helyett), ahol a szélesség és hosszúság interakcióját is bele foglaljuk a modellbe. Ekkor a koeficiensek és elojelek egy markáns változáson mennek keresztül. A hosszúság most pozitív koeficienssel, míg a szélesség negatív koeficiensse rendelkezik. A szélesség predikciós mértéke sem szignifikáns többé. Ugyanakkor a long, lat és long:lat változók VIF értéke rendkívül magas, jelentős multikollinearitást jelezve. Az eredményeinkben látható jelentős változás a kollinearitás miatti instabilitásból fakad. Bár a modell koeficiensei jelentősen megváltoztak, a modell R^2 értéke csak egész kicsit mozdult el.

```
mod_house_geolocation_inter = lm(price_mill_HUF ~ sqm_living +
  grade + long * lat, data = data_house_nooutliers)

summary(mod_house_geolocation_inter)
```

```
##
```



```
## Call:
## lm(formula = price_mill_HUF ~ sqm_living + grade + long * lat,
##     data = data_house_nooutliers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -65.679 -22.234  -4.974  17.289 140.811
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.461e+06  7.802e+05   1.873  0.0626 .
## sqm_living    3.814e-01  5.602e-02   6.809 1.23e-10 ***
## grade         1.597e+01  3.385e+00   4.717 4.59e-06 ***
## long          1.202e+04  6.384e+03   1.882  0.0613 .
## lat          -3.079e+04  1.641e+04  -1.876  0.0621 .
## long:lat      -2.532e+02  1.342e+02  -1.886  0.0608 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.3 on 192 degrees of freedom
## Multiple R-squared:  0.5934, Adjusted R-squared:  0.5828
## F-statistic: 56.03 on 5 and 192 DF,  p-value: < 2.2e-16

mod_house_geolocation_inter %>%
  vif()
```

```
##      sqm_living      grade      long      lat      long:lat
## 1.885276e+00 1.872074e+00 1.799568e+05 8.275735e+05 1.125795e+06
```

A long:lat interakciós taggal a long és a lat prediktorok erős korrelációt mutatnak. Itt szerkezeti kollinearitásról beszélhetünk, hiszen a multikollinearitást két már a modellünkben lévő prediktorból kepezett új prediktor okozza. A kevert interakciós tag mind a long mind pedig a lat prediktoroktól függ, ebből származik a magas korreláció.

A változók standardizálásával megoldható a probléma. Egy lehetséges **jó megoldás a “centrálás”**, azaz minden érintett prediktor esetén a mintaátlagot kivonjuk az egyes értékekből. Ezzel a módszerrel megőrizzük a változók eredeti skáláját, és így, a egyutthatok ugyan azt fogják jelenteni mint korábban, a centrálás előtt. (Használhatnánk Z transzformációt is, de az a egyutthatok értelmezését is megváltoztatná, mivel ilyenkor a prediktorok skálája is változik.)

A longitude (hosszúság) és latitude (szélesség) centrálását követően a kollinearitás megszűnik, és megközelítőleg hasonló a hatás mértéket is mint az interakció bevonása előtt. A szélességnek megint van szignifikáns prediktív értéke. Ugyanakkor az R^2 érték teljesen érintetlenül marad a multikollinearitás eltávolítása során. Azaz az előrejelzésekre való alkalmasság változatlanul marad, de a regressziós egyutthatok stabilabbá és könnyebben interpretálhatóvá váltak.

```
data_house_nooutliers = data_house_nooutliers %>%
  mutate(long_centered = long - mean(long), lat_centered = lat -
    mean(lat))

mod_house_geolocation_inter_centered = lm(price_mill_HUF ~ sqm_living +
  grade + long_centered * lat_centered, data = data_house_nooutliers)

mod_house_geolocation_inter_centered %>%
  vif()
```

```
##           sqm_living           grade
##           1.885276           1.872074
##           long_centered       lat_centered
##           1.058954           1.049776
## long_centered:lat_centered
##           1.050759

summary(mod_house_geolocation_inter_centered)

##
## Call:
## lm(formula = price_mill_HUF ~ sqm_living + grade + long_centered *
##     lat_centered, data = data_house_nooutliers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -65.679 -22.234  -4.974   17.289  140.811
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -49.00975    20.17976   -2.429   0.0161 *
## sqm_living         0.38144     0.05602    6.809 1.23e-10 ***
## grade            15.96840     3.38501    4.717 4.59e-06 ***
## long_centered    -25.83501    15.48572   -1.668   0.0969 .
## lat_centered     151.80613    18.47973    8.215 3.08e-14 ***
## long_centered:lat_centered -253.17679   134.24659   -1.886   0.0608 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.3 on 192 degrees of freedom
## Multiple R-squared:  0.5934, Adjusted R-squared:  0.5828
## F-statistic: 56.03 on 5 and 192 DF,  p-value: < 2.2e-16
```

5.4.2.2 Adat kollinearitás kezelése Állítsunk elő egy modellt, melyben az `sqm_living` és `grade` változókon túl az `sqm_above` változót is használjuk mint prediktort (az ingatlan földfelszín feletti területe négyzetmeterben).

A vif itt 3 feletti.

```
mod_house5 = lm(price_mill_HUF ~ sqm_living + grade + sqm_above,
  data = data_house)

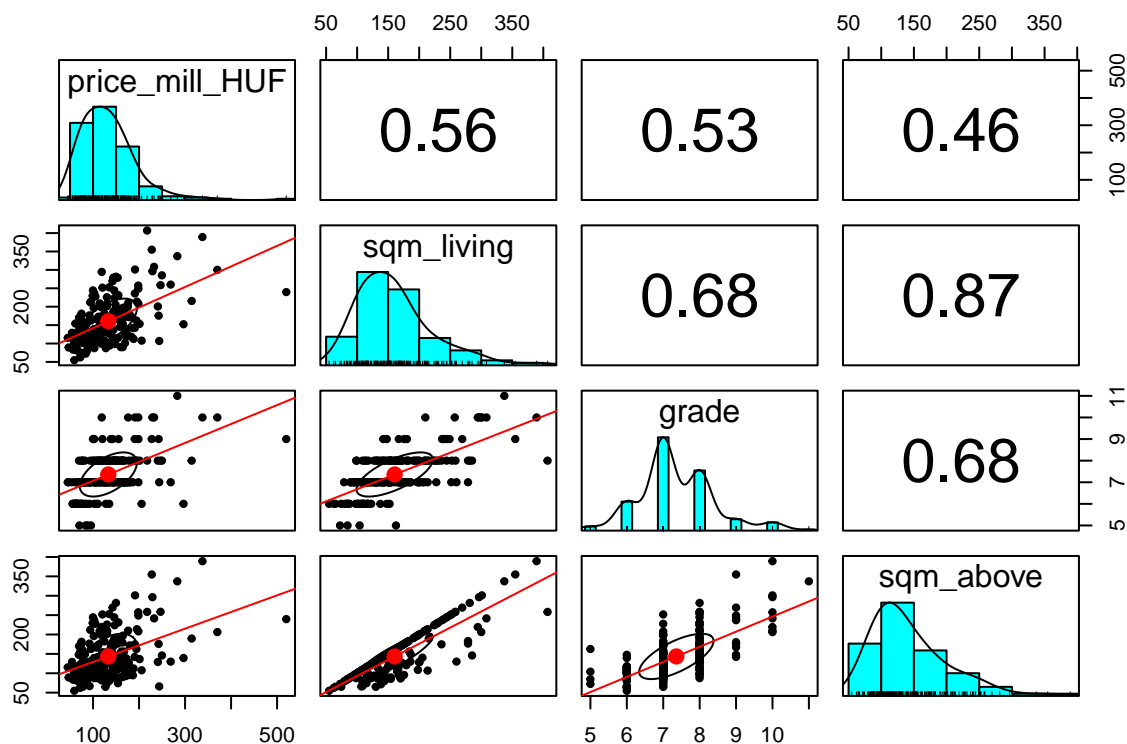
vif(mod_house5)
```

```
## sqm_living    grade  sqm_above
##   4.505215    1.972816   4.532763
```

Ennek okát megvizsgálhatjuk a prediktorok korrelációs mátrixának tanulmányozásával. A `pairs.panels()` függvény sok hasznos diagrammal szolgál, melyeken nem csak a változók közti korrelációt, de a változók eloszlását, illetve a páronkénti kapcsolatuk scatter plotjait is láthatjuk.

A korrelációs mátrix alapján egyértelmű, hogy az `sqm_living` és `sqm_above` korrelációja igen magas.

```
data_house %>%
  select(price_mill_HUF, sqm_living, grade, sqm_above) %>%
  pairs.panels(col = "red", lm = T)
```



A ket modell eredményenek összehasonlítása alapján mit állapíthatunk meg? Ertelmezd a ket modell regressziós együtthatóit!

```
summary(mod_house5)
```

```
##
## Call:
## lm(formula = price_mill_HUF ~ sqm_living + grade + sqm_above,
##     data = data_house)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -122.62  -26.66   -7.43   20.90  336.08
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -63.2813    28.1906  -2.245   0.0259 *
## sqm_living      0.5880     0.1208   4.868 2.32e-06 ***
## grade          19.5424     4.8139   4.060 7.10e-05 ***
## sqm_above     -0.2905     0.1275  -2.279   0.0238 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.44 on 196 degrees of freedom
## Multiple R-squared:  0.3746, Adjusted R-squared:  0.365
## F-statistic: 39.13 on 3 and 196 DF, p-value: < 2.2e-16
```

```
summary(mod_house3)
```

```
##
## Call:
## lm(formula = price_mill_HUF ~ sqm_living + grade, data = data_house_nooutliers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.130  -28.175   -4.994   21.582  154.004
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -48.21536    23.84403  -2.022   0.0445 *
## sqm_living    0.34451     0.06614   5.208 4.82e-07 ***
## grade        16.78639     4.01091   4.185 4.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.08 on 195 degrees of freedom
## Multiple R-squared:  0.413, Adjusted R-squared:  0.407
## F-statistic: 68.59 on 2 and 195 DF, p-value: < 2.2e-16
```

A multikollinearitásból fakadóan nem bízhatunk a mod_house5 egyes együtthatóiban, sem a prediktorokra vonatkozó t-próbákban.

Több lehetőség is nyitva áll a kollinearitás megoldására:

1. A szorosan korreláló prediktorok valamelyikének **eltávolítása**,
2. A prediktorok **lineáris kombinálása**, pl minden megfigyeléshez két adat **átlagát** használni. (pl a 3as esetben sqm_living = 1050, sqm_above = 950, azaz az átlag ebben az esetben 1000). Ugyanakkor az nem egyértelmű, hogy hogyan értelmeznénk az egyes prediktorokat.
3. Használhatóak akár teljesen más statisztikai eljárások is (pl.: parciális legkisebb négyzetek regressziója, vagy **fokkomponens elemzés**)

Jelenleg a legkezenfekvőbb talán az első módszer, vagyis a két problémás prediktor közül az egyik eltávolítása a modellből, hiszen az sqm_living és sqm_above nagyon megegyeznek és konceptuálisan sem igazan hordoznak különbozó információt. Válasszuk ki azt, amelyik intuitíve többet számít az ingatlan árába! Jelen esetben talán az sqm_living megtartása lehet célravezető, hiszen az az elméletünk, hogy a lakható terület mérete befolyásolja az ingatlan árát, a pince meglétérol szolgáltató információt pedig figyelembe vehetjük a has_basement változóval is egy későbbi modellben. ha van valamiféle ismeretünk korábbi, ingatlan árazást érinto kutatásokról, úgy azt is felhasználhatjuk annak eldöntésében, hogy melyik prediktort érdemes elhagyni. Praktikus szempontokat is figyelembe vehetünk, mint például azt, hogy az össz lakóterület könnyebben hozzáférhető információ, mint a földfelszín feletti terület, így ha az össz lakóterületet választjuk ki mint prediktort, akkor több esetben használhatjuk modellünket az árak előrejelzésére.

Azt, hogy melyik prediktort hagyjuk fel, és melyiket tartjuk meg, elméleti alapon, vagy korábbi kutatási eredmények alapján kell eldönteni, ezért nem javasolt hogy ezt a döntést a modellek illeszkedésének összehasonlítása alapján hozzuk meg.

További anyagok az alábbi linkeken:

<https://statisticalhorizons.com/multicollinearity>

<http://blog.minitab.com/blog/understanding-statistics/handling-multicollinearity-in-regression-analysis>

<http://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/>

Gyakorlás

Végezzünk modell diagnosztikát a ma tanultak alapján, egy új lineáris modellen, ahol az “sqm_living”, “sqm_living15”, “yr_built”, és “condition” prediktorok alapján határozzuk meg az ingatlan árakat.
