Zmienne kategorialne w modelu liniowym

Bartosz Maćkiewicz

Wstęp

Dzisiejsze zajęcia poświęcone będą zmiennym kategorialnym w modelu liniowym. Do tej pory koncentrowaliśmy się na zmiennych liczbowych i relacjach między nimi. Teraz jednak rozszerzymy naszą wiedzę i zobaczymy, że możemy wykorzystać model liniowy również wówczas, gdy nasze zmienne objaśniające są są zmiennymi nominalnymi (kategorialnymi). Przykładem takich zmiennych może być płeć, wykształcenie, preferencje polityczne, religia, ale także – uwaga – warunek eksperymentalny, w którym znalazł się uczestnik badania (np. rodzaj terapii) albo rodzaj próby w eksperymencie.

Na początek wczytamy dane dotyczące *guilty pleasure* z eksperymentu omawianego na zajęciach. Następnie zakodujemy sobie od razu poszczególne wyniki tak, jak robiliśmy to na pierwszych zajęciach poświęconych regresji.

(W dzisiejszym notatniku celowo nie użyjemy pakietu `knitr` oraz `pander` aby opatrzyli się Państwó również ze standardowum *outputem* R.)

```
data <- read.csv("Study3_Results_AfterExclusion.csv", header = T)

data$ART <- apply(data[,c("ART1", "ART2", "ART3")], 1, mean)
data$IDEAL <- apply(data[,c("IDEAL1", "IDEAL2")], 1, mean)
data$PEOPLE <- apply(data[,c("PEOPLE1", "PEOPLE2")], 1, mean)
data$SOCIAL <- apply(data[,c("SOCIAL1", "SOCIAL2")], 1, mean)
data$INTELLECTUAL <- apply(data[,c("INTELLECTUAL1", "INTELLECTUAL2")], 1, mean)
```

Do tej pory wszystko co tutaj zrobiliśmy powinno być dla Państwa zrozumiałe. Obliczyliśmy średnią z pytań `ART1`, `ART2` i `ART3` dla każdego badanego, a następnie zapisaliśmy ją w kolumnie `ART`. Tak samo postąpiliśmy z innymi pytaniami.

Kontrasty proste

Załóżmy, że chcielibyśmy dowiedzieć się, czy kobiety są bardziej skłonne od mężczyzn do wyższych odpowiedzi na pytania testujące wyjaśnienie odwołujące się do wartości intelektualnych dzieła (tzn. np. "This work is not very complex or intellectual"). Załóżmy, że mamy hipotezę, która głosi, że kobiety są przemądrzałe i będą prezentować wyższy poziom zgody na takie stwierdzenia dotyczące przyczyn ich *guilty pleasure*.

Na początek odfiltrujmy tylko te obserwacje, w których mamy jakąkolwiek wartość w kolumnie `GENDER`.

```
data_filtered <- data[data$GENDER != "",]
# usuwamy z naszego factora nieużywany, pusty poziom
# uwaga!
```

Następnie stwórzmy model, w którym za pomocą zmiennej `GENDER` przewidywać będziemy wynik na skali `INTELLECTUAL`.

```
fit <- lm(INTELLECTUAL ~ GENDER, data = data_filtered)
summary(fit)
```

```
##
```

```
## Call:
## lm(formula = INTELLECTUAL ~ GENDER, data = data_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9151 -0.8557  0.0849  1.0849  2.7037
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.2963     0.2062  20.834   <2e-16 ***
## GENDERA woman   0.6188     0.2930   2.112   0.0371 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.515 on 105 degrees of freedom
## Multiple R-squared:  0.04074,    Adjusted R-squared:  0.03161
## F-statistic:  4.46 on 1 and 105 DF,  p-value: 0.03707
```

Jak widzimy `GENDER` osiągnęło tutaj próg statystycznej istotności dla $\alpha = 0.05$.

W Państwa głowach w tym momencie powinno pojawić się pytanie. "Dlaczego nie wykorzystaliśmy po prostu testu t Studenta, tak jak to robiliśmy do tej pory?" - mogą Państwo zapytać. Pytanie jest oczywiście zasadne. Zobaczmy, czy test t Studenta da nam inny wynik niż analiza regresji:

```
t.test(INTELLECTUAL ~ GENDER, var.equal = T, data = data_filtered)
```

```
##
##  Two Sample t-test
##
## data:  INTELLECTUAL by GENDER
## t = -2.1119, df = 105, p-value = 0.03707
## alternative hypothesis: true difference in means between group A man and group A woman is not equal t
## 95 percent confidence interval:
##  -1.19978685 -0.03780923
## sample estimates:
##   mean in group A man mean in group A woman
##              4.296296              4.915094
```

Okazuje się, że dla zmiennej kategorialnej posiadającej dwa poziomy (tutaj: mężczyzna i kobieta), test t Studenta i analiza regresji są sobie równoważne! Dodatkowo, dzięki spojrzeniu na wynik testu t Studenta widzimy bardzo dobrze jak interpretować w tym wypadku wyraz wolny regresji oraz współczynniki regresji.

Wyraz wolny regresji (w tabeli: `(Intercept)`) wynosił w tym wypadku 4.2963. Patrząc na wynik testu t-Studenta widzimy, że jest to po prostu średnia `INTELLECTUAL` dla grupy mężczyzn! Współczynnik regresji dla `GENDER: A woman` wynosi 0.6188 i jest to po prostu różnica między średnią `INTELLECTUAL` u kobiet i u mężczyzn.

Oczywiście w przypadku regresji mamy znacznie większe możliwości niż przy teście t Studenta. Możemy, dodając więcej zmiennych różnego rodzaju (liczbowych i nominalnych) kontrolować wpływ tych czynników na interesującą nas różnicę.

Fakt, że jakaś znana już Państwu procedura będzie równoważna modelowi liniowemu, będzie powracającym motywem na tych zajęciach.

Zobaczmy co się stanie w przypadku, gdy nasza zmienna będzie miała więcej niż dwa poziomy. Badacze pytali również o to, z jakim innym odczuciem łączy się *guilty pleasure* odczuwane przez badanych (kolumna `EMOTION`). Badani mogli wybrać jedną z pięciu opcji (obrzydzenie, wina, zażenowanie, zawstydzenie, smutek) lub wpisać własne.

Zobaczmy, czy to, które uczucie wskazywali, wpływa na ich wynik na skali `PEOPLE` (intersubiektywna zgoda). Na początek odfiltrujemy tylko te obserwacje, w których badani udzielili jednej z predefiniowanych odpowiedzi.

```
ls <- c("DISGUST", "GUILT", "EMBARRASSMENT", "SHAME", "SADNESS")
data_filtered <- data[data$EMOTION %in% ls, ]
```

Następnie stwórzmy odpowiedni model i przyjrzyjmy się mu dokładnie:

```
fit <- lm(PEOPLE ~ EMOTION, data = data_filtered)
summary(fit)
```

```
##
## Call:
## lm(formula = PEOPLE ~ EMOTION, data = data_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6410 -0.6410 -0.0625  0.8958  2.8590
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.56250    0.46730   7.624 1.45e-11 ***
## EMOTIONEMBARRASSMENT  1.54167    0.50474   3.054  0.00289 **
## EMOTIONGUILT          0.07853    0.51299   0.153  0.87865
## EMOTIONSADNESS        0.31250    0.80938   0.386  0.70025
## EMOTIONSHAME          1.10417    0.71381   1.547  0.12505
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.322 on 100 degrees of freedom
## Multiple R-squared:  0.2339, Adjusted R-squared:  0.2032
## F-statistic: 7.631 on 4 and 100 DF,  p-value: 2.086e-05
```

Widzimy, że w naszym modelu mamy cztery czynniki (zasadniczo uniwersalnym faktem jest to, że będziemy ich mieli $k-1$, gdzie k to liczba poziomów naszej zmiennej kategorialnej). Jak łatwo zauważyć, brakuje nam obrzydzenia (`DISGUST`). Oznacza to, że średni wynik dla `DISGUST` jest właśnie naszą wartością referencyjną, do której porównujemy pozostałe średnie. Łatwo możemy się o tym przekonać, zobaczywszy średnie dla poszczególnych grup (proszę zwrócić uwagę - `DISGUST` ma średnią 3.56, co jest w tym wypadku równe wyrazowi wolnemu regresji.

```
tapply(data_filtered$PEOPLE, data_filtered$EMOTION, mean)
```

```
##       DISGUST EMBARRASSMENT         GUILT       SADNESS         SHAME
##      3.562500      5.104167      3.641026      3.875000      4.666667
```

Co jednak gdybyśmy chcieli wybrać inną grupę jako grupę, do której porównujemy pozostałe? W takim wypadku musimy ręcznie ustawić odpowiednie kontrasty. W przypadku kontrastów prostych (a takimi się teraz zajmowaliśmy nawet o tym nie wiedząc!) możemy zrobić to używając funkcji `contr.treatment`.

Przy wyborze grupy referencyjnej powinniśmy kierować się kilkoma zasadami.

Po pierwsze, porównania do grupy referencyjnej powinny być użyteczne. Jeśli na przykład porównujemy ze sobą jakieś terapie, to warto jako jako grupę referencyjną potraktować grupę kontrolną, gdzie uczestnicy nie dostawali żadnej terapii lub – jeżeli takiej grupy w badaniu nie było – grupę, która przyjmowała „standardową'' terapię, której skuteczność powinna być najniższa (w porównaniu do fantastycznych nowych terapii, które testujemy!).

Po drugie, nie warto jako referencyjnych wybierać takich poziomów jak "Inne" albo "Nie określono", ponieważ wtedy nasze porównania nie będą jasne (nie bardzo wiemy, do czego porównujemy, skoro jako referencyjną

wybraliśmy „śmietnikową'' (*waste-basket*) kategorię).

Po trzecie, warto wybrać grupę, w której mamy relatywnie dużo obserwacji (w porównaniu do pozostałych grup). Dzięki temu nasze wyniki będą lepiej replikowalne.

Wróćmy teraz do naszej zmiennej `EMOTION`. W tym wypadku trudno zastosować pierwszą i drugą zasadę, więc skoncentrujemy się na trzeciej. Najwięcej obserwacji mamy w grupie, która wskazała na `EMBARASSMENT`, dlatego to właśnie ją uczynimy grupą referencyjną.

```
data_filtered$EMOTION <- factor(data_filtered$EMOTION)
contrasts(data_filtered$EMOTION) <- contr.treatment(5, # sumaryczna liczba poziomów zmiennej
                                                    base = 2 # który poziom ma być referencyjny
                                                    )
```

Teraz możemy jeszcze raz powtórzyć naszą regresję. Tym razem porównywać będziemy do grupy, która wskazała na emocję "zażenowanie" (`EMBARASSMENT`).

```
fit <- lm(PEOPLE ~ EMOTION, data = data_filtered)
summary(fit)
```

```
##
## Call:
## lm(formula = PEOPLE ~ EMOTION, data = data_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6410 -0.6410 -0.0625  0.8958  2.8590
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.1042     0.1908  26.755  < 2e-16 ***
## EMOTION1     -1.5417     0.5047  -3.054  0.00289 **
## EMOTION3     -1.4631     0.2849  -5.135 1.39e-06 ***
## EMOTION4     -1.2292     0.6878  -1.787  0.07697 .
## EMOTION5     -0.4375     0.5723  -0.764  0.44641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.322 on 100 degrees of freedom
## Multiple R-squared:  0.2339, Adjusted R-squared:  0.2032
## F-statistic: 7.631 on 4 and 100 DF,  p-value: 2.086e-05
```

Jak widzimy zmienił się nasz wyraz wolny regresji, współczynniki regresji oraz wartośc p. Nic dziwnego! Teraz naszej grupy odniesienia nie wyznacza już poziom `DISGUST`, ale `EMBARASSMENT`.

Być może jednak takie porównania nas nie interesują. Nie zawsze jest bowiem sens porównywać średnie dla poszczególnych grup do jakiejś jednej, z góry ustalonej grupy. Czasami chcielibyśmy się dowiedzieć czegoś innego. Z pomocą nam przyjdą tutaj inne kontrasty. Pierwszym rodzajem, którym się zajmiemy, są kontrasty odchylenia.

Zanim jednak przejdziemy do innych rodzajów kontrastów, spróbujmy zobaczyć co tak naprawdę robi funkcja `contr.treatment`, której użyliśmy przed chwilą.

```
contr.treatment(5)
```

```
##   2 3 4 5
## 1 0 0 0 0
## 2 1 0 0 0
## 3 0 1 0 0
```

```
## 4 0 0 1 0
## 5 0 0 0 1
```

Funkcja ta zwraca nam macierz kontrastów. Możemy o tym myśleć jako o sposobie kodowania naszej zmiennej kategorialnej. 5 poziomów zmiennej kategorialnej zakodowaliśmy za pomocą czterech binarnych zmiennych. Pierwszy poziom kodujemy za pomocą wektora $(0, 0, 0, 0)$, drugi to $(1, 0, 0, 0)$, trzeci to $(0, 1, 0, 0)$, czwarty to $(0, 0, 1, 0)$ i ostatni piąty to $(0, 0, 0, 1)$. Schemat ten dość łatwo zrozumieć i jest na pierwszy rzut oka naturalny.

Każdy element takiego wektora opisuje dychotomiczną cechę obserwacji - należenie bądź nienależenie do jakiejś kategorii. Często spotkają Państwo określenie *dummy coding*. Inne schematy kodowania będą wyglądać podobnie, ale różnić się będą tym, jak należy analizę, w której są one użyte, interpretować.

Przyjrzyjmy się jeszcze równaniu regresji w przypadku *dummy coding*. Dla omawianego przez nas przykładu pięciu poziomów zmiennej kategorialnej będzie ono miało poznać:

$$\hat{Y} = B_1C_1 + B_2C_2 + B_3C_3 + B_4C_4 + B_0$$

Załóżmy więc, że mamy 5 kategorii: `DISGUST` (C_1), `EMBARRASSMENT`, `GUILT` (C_2), `SADNESS` (C_3) oraz `SHAME` (C_4) a kategorią referencyjną jest `EMBARASSMENT`. W *dummy coding* grupa referencyjna zakodowana jest jako same zera, dlatego dla tej grupy równanie możemy zredukować:

$$\hat{Y} = B_1(0) + B_2(0) + B_3(0) + B_4(0) + B_0 = B_0 = M_{Embarassment}$$

Widzimy więc teraz, dlaczego przy *dummy coding* wyraz wolny regresji to średnia dla grupy referencyjnej. Przeanalizujmy jeszcze jeden przykład, tym razem obserwacji z grupy `GUILT`.

$$\hat{Y} = B_1(0) + B_2(1) + B_3(0) + B_4(0) + B_0 = B_0 + B_2 = M_{Guilt}$$

Teraz już rozumiemy, dlaczego współczynnik regresji dla `GUILT` był różnicą między średnią dla grupy referencyjnej a średnią dla grupy `GUILT`!

Pamiętajmy jednak, że niezależnie od tego, jak zakodujemy nasze zmienne nominalne na użytek regresjii liniowej, kilka rzeczy się nie zmienia. Będziemy mieli taką samą wartość R^2 i taką samą związaną z nim wartość p. Nie dodajemy ani nie odejmujemy żadnych informacji, decydując się na inny schemat kodowania, dlatego nie powinno nas to specjalnie tutaj dziwić. Inny sposób kodowania będzie jednak zmieniał interpretacje wyrazu wolnego oraz poszczególnych współczynników B dla zmiennych uwzględnionych w regresji.

(Uwaga! Standaryzowane współczynniki regresji dla zmiennych nominalnych nie są specjalnie użyteczne, ponieważ ich odchylenie standardowe zależy od tego, ile obserwacji jest w danej kategorii! Proszę o tym pamiętać! I zastanowić się dlaczego tak jest!)

To, który schemat kodowania wybierzmy, zależy od tego na jakie pytania badawcze chcemy odpowiedzieć za pomocą przeprowadzanej przez nas analizy.

Kontrasty odchylenia

Kontrasty odchylenia pozwalają nam porównywać średnie dla poszczególnych poziomów naszej zmiennej kategorialnej nie do jednego, referencyjnego poziomu, ale do ogólnej średniej ze średnich dla wszystkich poziomów naszej zmiennej.

W przypadku zmiennej `EMOTION` takie porównanie może być bardziej przydatne - nie mamy w tym wypadku powodu, dla którego jakichś jeden jej poziom miałby mieć szczególne znaczenie. Możemy powiedzieć R-owi, że chcemu użyć kontrastów odchylenia korzystając z funkcji `contr.sum` (proszę dla zaspokojenia ciekawości zobaczyć sobie samemu jak wygląda macierz zwracana przez tę funkcję).

```r
contrasts(data_filtered$EMOTION) <- contr.sum(5)
fit <- lm(PEOPLE ~ EMOTION, data = data_filtered)
summary(fit)
```

```
##
## Call:
## lm(formula = PEOPLE ~ EMOTION, data = data_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6410 -0.6410 -0.0625  0.8958  2.8590
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.1699     0.2027  20.569  < 2e-16 ***
## EMOTION1     -0.6074     0.4149  -1.464 0.146330
## EMOTION2      0.9343     0.2509   3.724 0.000324 ***
## EMOTION3     -0.5288     0.2607  -2.028 0.045175 *
## EMOTION4     -0.2949     0.5506  -0.536 0.593447
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.322 on 100 degrees of freedom
## Multiple R-squared:  0.2339, Adjusted R-squared:  0.2032
## F-statistic: 7.631 on 4 and 100 DF,  p-value: 2.086e-05
```

W tym wypadku wyraz wolny regresji to średnia ze średnich obliczonych dla każdego poziomu naszego czynnika. Możemy to potwierdzić obliczając ręcznie średnią ze średnich dla wszystkich poziomów naszej zmiennej.

```r
mean(tapply(data_filtered$PEOPLE, data_filtered$EMOTION, mean))
```

```
## [1] 4.169872
```

Współczynniki regresji dla poszczególnych poziomów naszej zmiennej są w tym wypadku różnicami między średnimi dla danego poziomu i średnią ze średnich dla wszystkich poziomów. Możemy się o tym łatwo przekonać.

```r
srednie <- tapply(data_filtered$PEOPLE, data_filtered$EMOTION, mean)
srednia_srednich <- mean(srednie)
srednie - srednia_srednich
```

```
##       DISGUST EMBARRASSMENT         GUILT       SADNESS         SHAME
##    -0.6073718     0.9342949    -0.5288462    -0.2948718     0.4967949
```

Potencjalnym problemem może być jednak to, że mamy podanego współczynnika regresji oraz wyniku testu statystycznego dla jednego z poziomów naszej zmiennej. Co do zasady powinniśmy się z tym pogodzić. Tutaj, w przeciwieństwie do *dummy coding* jako grupę bazową powinniśmy wybrać tę, która jest dla nas z punktu widzenia stawianych przez nas pytań badawczych najmniej interesująca. Możemy z tym sobie (dość nieelegancko) poradzić przestawiając poziomy zmiennej w taki sposób, żeby inny poziom był wyłączony z modelu.

```r
data_filtered$EMOTION_ALT <- factor(data_filtered$EMOTION, levels = ls)
```

Teraz możemy jeszcze raz przeprowadzić regresję liniową. Proszę zwrócić uwagę na fakt, że współczynniki regresji nie zmieniły się, chociaż na pierwszy rzut oka mamy inny zestaw predyktorów.

```
contrasts(data_filtered$EMOTION_ALT) <- contr.sum(5)
fit <- lm(PEOPLE ~ EMOTION_ALT, data = data_filtered)
summary(fit)
```

```
##
## Call:
## lm(formula = PEOPLE ~ EMOTION_ALT, data = data_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6410 -0.6410 -0.0625  0.8958  2.8590
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.1699     0.2027  20.569  < 2e-16 ***
## EMOTION_ALT1  -0.6074     0.4149  -1.464 0.146330
## EMOTION_ALT2  -0.5288     0.2607  -2.028 0.045175 *
## EMOTION_ALT3   0.9343     0.2509   3.724 0.000324 ***
## EMOTION_ALT4   0.4968     0.4645   1.069 0.287441
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.322 on 100 degrees of freedom
## Multiple R-squared:  0.2339, Adjusted R-squared:  0.2032
## F-statistic: 7.631 on 4 and 100 DF,  p-value: 2.086e-05
```

Dlaczego teraz nasze współczynniki regresji oraz wyraz wolny mają inną interpretację, niż w wypadku *dummy coding*? Zobaczmy jakie wartości przypisuje poszczególnym poziomem zmiennych funkcja `contr.sum`:

```
contr.sum(5)
```

```
##   [,1] [,2] [,3] [,4]
## 1    1    0    0    0
## 2    0    1    0    0
## 3    0    0    1    0
## 4    0    0    0    1
## 5   -1   -1   -1   -1
```

Widzimy, że jedyną różnicą w porównaniu do *dummy coding* jest to, że grupa referencyjna zakodowana jest za pomocą wartości -1 zamiast samych zer. Zmienia to jednak całkowicie interpretację wyrazu wolnego regresji oraz współczynników regresji, o czym należy pamiętać!

Kontrasty wielomianowe

Do tej pory poziomy naszej zmiennej kategorialnej nie były uporządkowe. Trudno sensownie wskazać jakiś porządek w sytuacji, kiedy na tapecie mamy pięć emocji albo dwie (lub więcej!) płcie. Co jednak z sytuacjami, gdy mamy zmienną na skali porządkowej? Przykładem takiej zmiennej z omawianego pytania jest `FREQUENCY`, które dotyczy częstotliwości występowania *guilty pleasure* u badanych. Przyjmuje ona pięć wartości – "Never", "Very rarely", "Sometimes", "Often" oraz "Very often". Załóżmy, że chcielibyśmy dowiedzieć się, czy istnieje związek między częstością występowania *guilty pleasure* a siłą przekonania o tym, że nie powinno się konsumować wywołujących *guilty pleasure* dzieł kultury. Możemy oczywiście odpowiednią zmienną uwzględnić w naszym modelu w domyślny dla R sposób:

```
fit <- lm(SHOULDNOT ~ FREQUENCY, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = SHOULDNOT ~ FREQUENCY, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7941 -1.2222  0.2059  1.2059  2.7778
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.000      1.685   1.780    0.078 .
## FREQUENCYOften           1.900      1.767   1.075    0.285
## FREQUENCYSometimes       1.794      1.698   1.057    0.293
## FREQUENCYVery often      2.333      1.946   1.199    0.233
## FREQUENCYVery rarely     1.222      1.716   0.712    0.478
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.685 on 104 degrees of freedom
## Multiple R-squared:  0.03619,    Adjusted R-squared:  -0.000882
## F-statistic: 0.9762 on 4 and 104 DF,  p-value: 0.4239
```

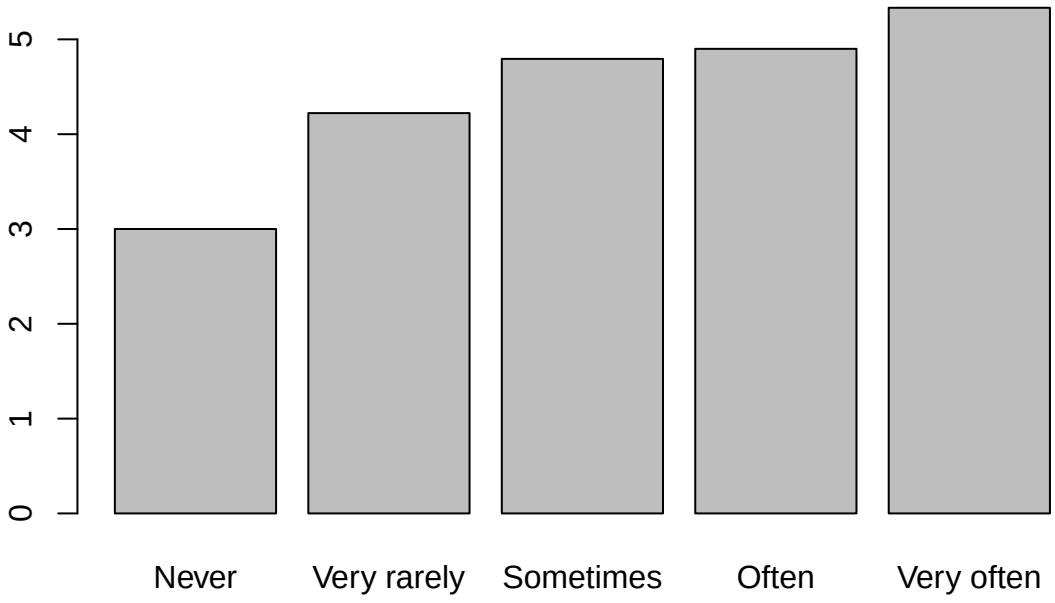
Interpretacja tak przeprowadzonej analizy nie jest jednak łatwa. R przyjął jako referencyjny poziom naszej zmiennej "Never" (pamiętajmy, domyślnie R używa kontrastów prostych!). Analiza ta nie pozwala nam to ocenić występowania żadnego trendu, ponieważ poziomy nie są uporządkowane. Spróbujmy ten problem rozwiązać. Na początek przekodujemy sobie naszą zmienną `FREQUENCY` tak, by miała typ `factor` i była uporządkowana (`ordered=T`).

```
data$FREQUENCY <- factor(data$FREQUENCY,
                         levels = c("Never", "Very rarely", "Sometimes", "Often", "Very often"),
                         ordered = T
                     )
```

Spróbujmy po przekodowaniu zmiennej `FREQUENCY` zilustrować średnie z `SHOULD` dla każdego z (już teraz uporządkowanych) poziomów powtórzyć naszą analizę.

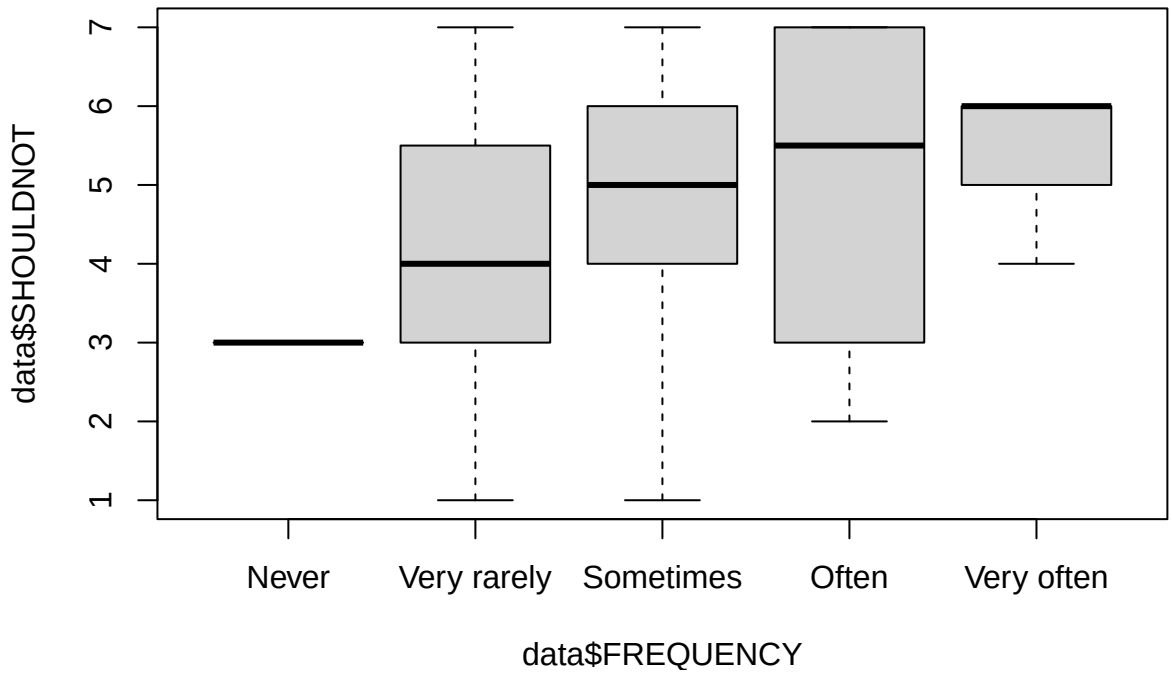
Na początek wykonajmy wykres słupkowy.

```
barplot(tapply(data$SHOULDNOT, data$FREQUENCY, mean))
```



Następnie zróbmy wykres pudełkowy.

```
plot(data$SHOULDNOT ~ data$FREQUENCY)
```



Widzimy, że w naszych danych widoczny jest pewien trend - im częściej ktoś doświadczał *guilty pleasure* tym silniej zgadza się, że nie powinien go doświadczać.

Powtórzmy teraz naszą analizę.

```
fit <- lm(SHOULDNOT ~ FREQUENCY, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = SHOULDNOT ~ FREQUENCY, data = data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7941 -1.2222  0.2059  1.2059  2.7778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.44993    0.41073  10.834   <2e-16 ***
## FREQUENCY.L  1.69006    1.24641   1.356    0.178
## FREQUENCY.Q -0.54623    1.05906  -0.516    0.607
## FREQUENCY.C  0.30920    0.73097   0.423    0.673
## FREQUENCY^4  0.07281    0.40562   0.179    0.858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.685 on 104 degrees of freedom
## Multiple R-squared:  0.03619,    Adjusted R-squared:  -0.000882
## F-statistic: 0.9762 on 4 and 104 DF,  p-value: 0.4239
```

Jak widzimy, R po przekodowaniu danych do `factor` z uporządkowanymi poziomami, zrobił coś, czego nie robił wcześniej. Użył zamiast kontrastów prostych *kontrastów wielomianowych* i testował cztery rodzaje trendu - opisywanego przez funkcję liniową, kwadratową oraz wielomiany trzeciego i czwartego stopnia. Żaden z tych trendów nie okazał się jednak statystycznie istotny.

Żeby unaocznić jak działają kontrasty wielomianowe rozważmy najprostszy przypadek, to znaczy wielomian pierwszego stopnia czyli – w naszym wypadku – po prostu funkcję liniową. W kolumnie FREQUENCY_QUANT możemy znaleźć dane z kolumny FREQUENCY przekodowane na wartości liczbowe.

```
fit <- lm(SHOULDNOT ~ FREQUENCY_QUANT, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = SHOULDNOT ~ FREQUENCY_QUANT, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7106 -1.2910  0.2894  1.2894  2.7090
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.4519     0.6883   5.015 2.12e-06 ***
## FREQUENCY_QUANT   0.4196     0.2324   1.805   0.0738 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.667 on 107 degrees of freedom
## Multiple R-squared:  0.02956,    Adjusted R-squared:  0.02049
## F-statistic: 3.259 on 1 and 107 DF,  p-value: 0.07385
```

Domyślnie (z powodów matematycznych) `contr.poly` produkuje kontrasty wielomianowe $n-1$ stopnia gdzie n to liczba poziomów zmiennej. Możemy jednak dość łatwo zredukować liczbę kontrastów tak, aby testować tylko trend liniowy.

```
contrasts(data$FREQUENCY, 1) <- contr.poly(5)[,1:2]
fit <- lm(SHOULDNOT ~ FREQUENCY, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = SHOULDNOT ~ FREQUENCY, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7106 -1.2910  0.2894  1.2894  2.7090
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.7106     0.1621  29.065   <2e-16 ***
## FREQUENCY.L   1.3268     0.7350   1.805   0.0738 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.667 on 107 degrees of freedom
## Multiple R-squared:  0.02956,    Adjusted R-squared:  0.02049
## F-statistic: 3.259 on 1 and 107 DF,  p-value: 0.07385
```

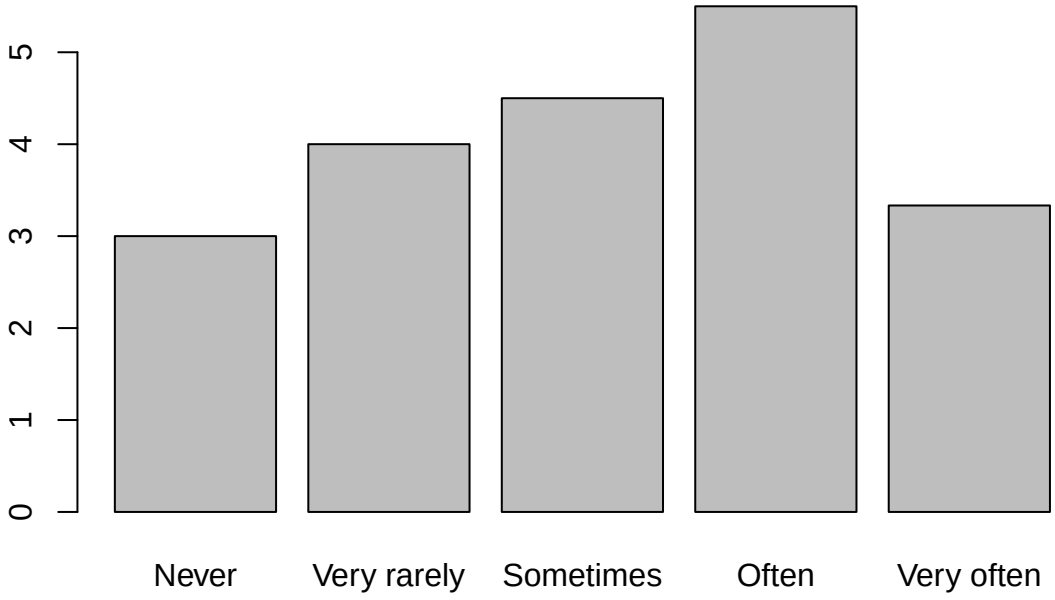
Jeśli to zrobimy, to zarówno wyraz wolny regresji, współczynnik regresji jak i wartość p będą takie same jak w przypadku, gdy użyliśmy zmiennej liczbowej (`FREQUENCY_QUANT`).

Odwrócone kontrasty Helmerta

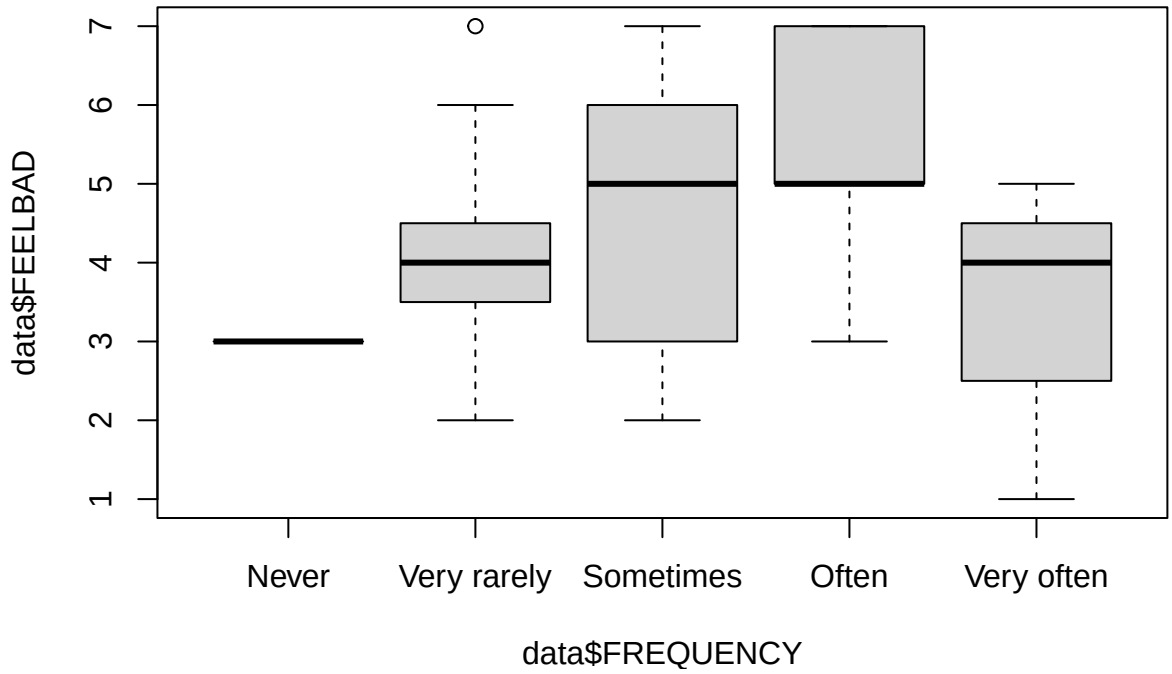
Korzystając z odwróconych kontrastów Helmerta dla każdego poziomu zmiennej kategorialnej (gdzie poziomy są uporządkowane, to znaczy mamy zmienną na skali porządkowej) porównujemy średnią dla danego poziomu ze średnią ze średnich dla wszystkich poziomów poprzednich. Innymi słowy, jeżeli mamy 4 poziomy jakiejś zmiennej, to porównujemy średnią dla poziomu 2 ze średnią dla poziomu 1, średnią dla poziomu 3 ze średnią ze średnich dla poziomów 1 i 2 oraz średnią dla poziomu 4 ze średnią ze średnich dla poziomów 1, 2 oraz 3.

Do czego taka analiza mogłaby się przydać? Rozważmy relację między tym, jak źle ktoś czuje się z poziomu *guilty pleasure* a tym, jak często go doświadcza. Na początek narysujmy sobie wykresy ilustrujące nasze dane.

```
barplot(tapply(data$FEELBAD, data$FREQUENCY, mean))
```



```
plot(data$FEELBAD ~ data$FREQUENCY)
```



Widzimy, że po pewnej tendencji wzrostowej ("Never" - "Very rarely" - "Sometimes" - "Often") następuje gwałtowny spadek. Im częściej ktoś odczuwa *guilty pleasure* tym gorzej się z tym czuje, chyba że odczuwa je bardzo często, wtedy jest mu wszystko jedno.

Spróbujmy powiedzieć R-owi żeby dla naszej zmiennej `FREQUENCY` skorzystał z odwróconych kontrastów Helmerta. Robimy to korzystając z funkcji `contr.helmert`.

```
contrasts(data$FREQUENCY) <- contr.helmert(5)
fit <- lm(FEELBAD ~ FREQUENCY, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = FEELBAD ~ FREQUENCY, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -2.5   -1.0    0.0    1.0    3.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.0667     0.3562  11.416   <2e-16 ***
## FREQUENCY1    0.5000     0.7442   0.672    0.503
## FREQUENCY2    0.3333     0.2550   1.307    0.194
## FREQUENCY3    0.4167     0.1702   2.449    0.016 *
## FREQUENCY4   -0.1833     0.1861  -0.985    0.327
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.462 on 104 degrees of freedom
## Multiple R-squared:  0.09167,    Adjusted R-squared:  0.05673
```

```
## F-statistic: 2.624 on 4 and 104 DF,  p-value: 0.03888
```

Wiersz `FREQUENCY 1` oznacza tutaj porównanie "Very rarely" z "Never", `FREQUENCY 2` – "Sometimes" z "Never" i "Very rarely", `FREQUENCY 3` – "Often" z "Sometimes", "Very rarely" oraz "Never" i w końcu `FREQUENCY 4` - "Very often" z resztą poziomów. Widzimy, że tylko dla trzeciego kontrastu osiągnęliśmy statystyczną istotność.

Kontrasty różnicowe

Czasami chcielibyśmy porównywać średnią dla danego poziomu nie ze średnią ze wszystkich poprzednich poziomów, ale ze średnią z poprzedniego poziomu. To umożliwią nam kontrasty różnicowe. W R możemy wygenerować takie konstrasty za pomocą funkcji `contr.sdif` z pakietu `MASS`.

```
contrasts(data$FREQUENCY) <- MASS::contr.sdif(5)
fit <- lm(FEELBAD ~ FREQUENCY, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = FEELBAD ~ FREQUENCY, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -2.5   -1.0    0.0    1.0    3.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.0667     0.3562  11.416   <2e-16 ***
## FREQUENCY2-1   1.0000     1.4884   0.672   0.5032
## FREQUENCY3-2   0.5000     0.3325   1.504   0.1356
## FREQUENCY4-3   1.0000     0.4950   2.020   0.0459 *
## FREQUENCY5-4  -2.1667     0.9621  -2.252   0.0264 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.462 on 104 degrees of freedom
## Multiple R-squared:  0.09167,    Adjusted R-squared:  0.05673
## F-statistic: 2.624 on 4 and 104 DF,  p-value: 0.03888
```

Łatwo możemy odczytać z tabeli, że statystycznie istotne są różnice między poziomami 4 a 3 (wzrost dla "Often" w porównaniu do "Sometimes") oraz między poziomami 5 i 4 (spadek dla "Very often" w porównaniu do "Often").