Wielokrotna (wieloraka) regresja liniowa

W tym odcinku kontynuować będziemy nasze przygody z wielokrotną regresją liniową. Tym razem wejdziemy troszkę głębiej w diagnostykę, założenia regresji oraz interpretację wyników. Cały czas korzystać będziemy z naszych danych dotyczących edukacji, które ściągnąć można z platformy. Na początek wczytajmy więc nasze dane i dla przypomnienia wyświetlmy sobie pięć pierwszych wierszy. Osoby, które nie pamiętają już czego dane dotyczą proszone są o zajrzenie do podręcznika i zajrzenie do notatnika z poprzedniego tygodnia.

```
library(pander)
library(knitr)
data <- read.table("Tab15-1.dat", header = T)
kable(head(data))
```

id	State	Expend	PTratio	Salary	PctSAT	Verbal	Math	SATcombined	PctACT	ACTcombined	LogPctSAT
1	Alabama	4.405	17.2	31.144	8	491	538	1029	61	20.2	2.079
2	Alaska	8.963	17.6	47.951	47	445	489	934	32	21.0	3.850
3	Arizona	4.778	19.3	32.175	27	448	496	944	27	21.1	3.296
4	Ark	4.459	17.1	28.934	6	482	523	1005	66	20.3	1.792
5	Calif	4.992	24.0	41.078	45	417	485	902	11	21.0	3.807
6	Col	5.443	18.4	34.571	29	462	518	980	62	21.5	3.367

Tolerancja i VIF

Współczynnik inflacji wariancji (*Variance Inflation Factor*) pozwala nam ocenić, czy predyktory są *współliniowe*. To, w jaki sposób ten współczynnik obliczyć nie jest w tej chwili istotne, potraktujmy go po prostu jako pewną wartość diagnostyczną. Jeżeli między predyktorami w naszym modelu zaobserwować możemy duży stopień współliniowości (powiedzmy, że $VIF > 4$), to wówczas może to skutkować niestabilnością naszego modelu. Tolerancja jest odwrotnością VIF i oblicza się ją $1/VIF$. Co do zasady dążymy do tego, aby tolerancja była jak największa a VIF jak najmniejszy. Mówi nam to, że nasz model charakteryzuje wysoka stabilność (niski błąd standardowy współczynników regresji). W przypadku $VIF > 10$ (silna współliniowość) powinniśmy dokonać korekty naszego modelu i usunąć ze zbioru predyktorów jedną zmienną.

VIF i Tolerancja pozwalają nam ocenić, które wybrane przez nas predyktory pokrywają się i w jakim stopniu.

W R VIF i tolerancje obliczyć może np. za pomocą funkcji `vif` z pakietu `car`. Stwórzmy model, w którym objaśniamy `SATcombined` za pomocą `Expend`, `LogPctSAT` oraz `PTratio`.

```
library(car)
```

```
## Ładowanie wymaganego pakietu: carData
```

```
fit <- lm(SATcombined ~ Expend + LogPctSAT + PTratio, data = data)
pander(summary(fit))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1132	39.79	28.45	7.644e-31
Expend	11.67	3.533	3.302	0.001863
LogPctSAT	-78.39	4.533	-17.29	9.344e-22
PTratio	0.7442	1.774	0.4194	0.6769

Table 3: Fitting linear model: SATcombined ~ Expend + LogPct-SAT + PTratio

Observations	Residual Std. Error	R^2	Adjusted R^2
50	26.01	0.8865	0.8791

Następnie obliczymy współczynnik inflacji wariancji (VIF).

```
pander(vif(fit))
```

Expend	LogPctSAT	PTratio
1.679	1.473	1.171

Oraz Tolerancję (podzielimy po prostu 1 przez uzyskane przez nas współczynniki VIF).

```
pander(1/vif(fit))
```

Expend	LogPctSAT	PTratio
0.5957	0.6787	0.8539

Jak widzimy VIF dla naszych predyktorów nie jest specjalnie wysoki, co uprawnia nas do stwierdzenie, że predyktory nie są ze sobą skorelowane w sposób zagrażający rzetelności naszego modelu.

Standaryzowane współczynniki regresji i „ważność'' poszczególnych predyktorów

Jednym z istotniejszych pytań, jakie zadać sobie możemy przeprowadzając analizę regresji, jest pytanie o to, które z wybranych przez nas predyktorów są najważniejsze. Pytanie to oczywiście interpretować można na wiele sposobów.

Błędem byłoby tutaj patrzeć po prostu na współczynnik regresji i uznać, że te predyktory, które mają wyższy współczynnik regresji, są ważniejsze. Jest to niewłaściwe rozumowanie, ponieważ współczynniki regresji zależą od tego w jakich jednostkach wyrażone są wartości naszych predyktorów. Możemy jednak sobie z tym poradzić standaryzując zmienne, których używamy w charakterze predyktorów (przypominam: odejmujemy od wszystkich wartości średnią i dzielimy wszystkie wartości przez odchylenie standardowe dzięki czemu nasze zmienne mają średnią 0 i odchylenie standardowe 1). Dodatkowo wystardaryzować możemy zmienną objaśnianą. Tak wystandaryzowane zmienne posłużyć nam mogą do przeprowadzenia regresji.

```
data$SATcombinedStd <- (data$SATcombined - mean(data$SATcombined))/sd(data$SATcombined)
data$ExpendStd <- (data$Expend - mean(data$Expend))/sd(data$Expend)
data$LogPctSATStd <- (data$LogPctSAT - mean(data$LogPctSAT))/sd(data$LogPctSAT)
data$PTratioStd <- (data$PTratio - mean(data$PTratio))/sd(data$PTratio)

fit_std <- lm(SATcombinedStd ~ ExpendStd + LogPctSATStd + PTratioStd, data = data)
pander(summary(fit_std))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.519e-16	0.04916	7.157e-15	1
ExpendStd	0.2125	0.06435	3.302	0.001863
LogPctSATStd	-1.042	0.06028	-17.29	9.344e-22
PTratioStd	0.02254	0.05375	0.4194	0.6769

Table 7: Fitting linear model: SATcombinedStd ~ ExpendStd + LogPctSATStd + PTratioStd

Observations	Residual Std. Error	R^2	Adjusted R^2
50	0.3476	0.8865	0.8791

Jak widzimy uzyskaliśmy trochę inne współczynniki regresji. Tym razem mają one postać standaryzowanych współczynników regresji. Warto zwrócić uwagę na to, że wyraz wolny regresji (*intercept*) wynosi w tym wypadku 0. Dlaczego? Dlatego, że wystandaryzowaliśmy zmienną objasnianą. Dodatkowo zmienia się teraz interpretacja wartości w kolumnie `Estimate` regresji - teraz, kiedy mamy tam standaryzowane współczynniki regresji, mówi nam ona, o ile odchyleń standardowych zmienia się nasza zmienna objaśniania wraz ze zmianą naszych predyktorów również wyrażonej w odchyleniach standardowych. Teraz, kiedy wszystko wyrazone jest już w odchyleniach standardowych, możemy sensownie porównywać predyktory między sobą.

Proszę zwrócić uwagę również na to, że standaryzacja zmiennych w regresji nie ma wpływu wartości statystyki testowej t ani na wartości p, chociaż oczywiście ma wpływ na błąd standardowy (bo zmieniliśmy jednostki!).

W zdefiniowanym przez nas sensie najważniejszym predyktorem jest `LogPctSAT` ponieważ jego wzrost o jedno odchylenie standardowe skutkuje zmianą w `SATcombined` wynoszącą -1.042 odchylenia standardowego.

To samo co zrobiliśmy „na piechotę'' w R możemy również zrobić za pomocą funkcji `lm.beta` z pakietu `lm.beta`. Jako argument przekazujemy nasz modela a funkcja ta sama uzupełni go o standaryzowane współczynniki regresji.

```
library(lm.beta)
pander(summary(lm.beta(fit)))
```

	Estimate	Standardized	Std. Error	t value	Pr(>|t|)
(Intercept)	1132	NA	39.79	28.45	7.644e-31
Expend	11.67	0.2125	3.533	3.302	0.001863
LogPctSAT	-78.39	-1.042	4.533	-17.29	9.344e-22
PTratio	0.7442	0.02254	1.774	0.4194	0.6769

Table 9: Fitting linear model: SATcombined ~ Expend + LogPctSAT + PTratio

Observations	Residual Std. Error	R^2	Adjusted R^2
50	26.01	0.8865	0.8791

Błędy standardowe współczynników regresji i testy statystyczne dla współczynników regresji

Przyjrzyjmy się ponownie stworzonemu przez nas modelowi.

```
pander(summary(lm.beta(fit)))
```

	Estimate	Standardized	Std. Error	t value	Pr(>|t|)
(Intercept)	1132	NA	39.79	28.45	7.644e-31
Expend	11.67	0.2125	3.533	3.302	0.001863
LogPctSAT	-78.39	-1.042	4.533	-17.29	9.344e-22

	Estimate	Standardized	Std. Error	t value	Pr(>|t|)
PTratio	0.7442	0.02254	1.774	0.4194	0.6769

Table 11: Fitting linear model: SATcombined ~ Expend + LogPct-SAT + PTratio

Observations	Residual Std. Error	R^2	Adjusted R^2
50	26.01	0.8865	0.8791

Widzimy, że w tabeli oprócz `Estimate` czyli współczynników regresji mamy również `Std.Error`, `t value` oraz `Pr(>|t|)`. Jak Państwo pamiętają `Pr(>|t|)` mówi nam, czy dany współczynnik regresji różni się w statystycznie istotny sposób od 0. R oblicza go za nas, ale nie jest trudno przeprowadzić taki test samemu. `Std.Error` jest błędem standardowym współczynnika kierunkowego regresji. Proszę mysleć o tej wartości tak samo, jak myślą Państwo o błędzie standardowym średniej – jest to (mówiąc trochę obrazowo) wartość odchylenia standardowego dla danej statystyki gdybyśmy brali taką samą próbę nieskończenie wiele razy i za każdym razem obliczali daną statystykę. Z konceptualnego punktu widzenia różnica jest taka, że w znanym Państwu przypadku było to odchylenie standardowe średnich z próby, tutaj jest to odchylenie standardowe współczynników kierunkowych regresji.

Statystykę testową t obliczyć możemy według wzoru:

$$t = \frac{b_j - b_j^*}{s_{b_j}}$$

gdzie b_j jest wartością naszego współczynnika kierunkowego regresji, b_j^* jest wartością przy założeniu hipotezy zerowej a s_{b_j} błędem standardowym. Dla $H0 : b_j^* = 0$ (czyli dla hipotezy zerowej głoszącej, że współczynnik kierunkowy regresji wynosi 0 czyli nie ma relacji między predyktorem a zmienną objaśnianą) wzór ten ma postać:

$$t = \frac{b_j}{s_{b_j}}$$

Rozkład statystyki testowej t przy założeniu hipotezy zerowej ma w tym wypadku $N - p - 1$ stopni swobody gdzie N to liczba obserwacji a p to liczba predyktorów.

Sprawdźmy to! Wiemy, że dla zmiennej `Expend` współczynnik kierunkowy wynosi 11.67 a błąd standardowy 3.533. Obliczmy więc że statystykę testową:

```
tstat = 11.67/3.533
tstat
```

```
## [1] 3.303142
```

Teraz możemy sprawdzić za pomocą funkcji `pt` jakie jest prawdopodobieństwo uzyskania takiej lub bardziej skrajnej wartości statystyki testowej:

```
pt(tstat, nrow(data) - 3 - 1, lower.tail = F) * 2 # mnożymy przez dwa bo dwustronny test
```

```
## [1] 0.001855948
```

Jak widzimy nasz wynik zgadza się z tym, który obliczył dla nas R. Uzyskaliśmy wartość p mniejszą niż 0.05, co uprawnia nas (przy założonym poziomie istotności statystycznej $\alpha = 0.05$) do odrzucenia hipotezy zerowej.

Założenia regresji liniowej

Jeżeli nasze predyktory są zmiennymi losowymi, to zakładamy, że łączny rozkład $Y, X_1, X_2, ..., X_p$ jest *wielowymiarowym rozkładem normalnym* (*multivariate normal distribution*).

Jeżeli poziomy naszych predyktorów są ustalone (czyli np. manipujemy nimi w ramach schematu eksperymentalnego), to wówczas musimy założyć, że rozkład warunkowy dla każdego poziomu predyktorów jest w naszej zmiennej objaśnianej Y normalny.

W praktyce jednak nawet dość spore odchylenia od tego założenia nie wpływają negatywnie na poziom błędu I rodzaju dla przeprowadzanych przez nas testów statystycznych. Warto jednak o tym pamiętać!

Na wikipedii można znaleźć ładne obrazki pokazujące jak wyglądają takie rozkłady. Zachęcam do pogooglowania (sam nie podejmuje się narysowania ich w R ponieważ są trójwymiarowe...)

Współczynnik R^2 i testy statystyczne

O R^2 mówiliśmy już sporo w poprzednich cześciach kursu, dlatego odsyłam do materiałów sprzed tygodnia i dwóch. Teraz jednak ponownie przyjrzyjmy się stworzonemu przez nas modelowi.

```
pander(summary(lm.beta(fit)))
```

	Estimate	Standardized	Std. Error	t value	Pr(>|t|)
(Intercept)	1132	NA	39.79	28.45	7.644e-31
Expend	11.67	0.2125	3.533	3.302	0.001863
LogPctSAT	-78.39	-1.042	4.533	-17.29	9.344e-22
PTratio	0.7442	0.02254	1.774	0.4194	0.6769

Table 13: Fitting linear model: SATcombined ~ Expend + LogPctSAT + PTratio

Observations	Residual Std. Error	R^2	Adjusted R^2
50	26.01	0.8865	0.8791

Widzimy, że R podaje nam dwie wartości statystyki R^2 - zwykłąd oraz „adjusted''. Wynika to z tego, że R^2 nie jest nieobciążonym estymatorem odpowiedniego parametru w populacji. Wersję nieobciążoną tego współczynnika oblicza się według wzoru:

$$R^2_{adj} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

gdzie N to liczba obserwacji a p to liczba predyktorów. Z jakiegoś powodu jednak w artykułach naukowych wartość ta pojawia się dość rzadko (tutaj zgadzam się z Howellem).

Możemy także przeprowadzić test statystyczny dla uzyskanego przez nas współczynnika R^2. R robi to za nas i możemy to zobaczyć jeśli wyświetlimy sobie pełne `summary` naszego modelu (pakiet `pander` trochę ucina, niestety).

```
summary(fit)
```

```
##
## Call:
## lm(formula = SATcombined ~ Expend + LogPctSAT + PTratio, data = data)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.153 -14.430  -2.416  18.020  56.068
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1131.9823    39.7887  28.450  < 2e-16 ***
## Expend        11.6651     3.5329   3.302  0.00186 **
## LogPctSAT    -78.3910     4.5332 -17.293  < 2e-16 ***
## PTratio        0.7442     1.7743   0.419  0.67687
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.01 on 46 degrees of freedom
## Multiple R-squared:  0.8865, Adjusted R-squared:  0.8791
## F-statistic: 119.8 on 3 and 46 DF,  p-value: < 2.2e-16
```

Naszą hipoteza zerowa głosi w tym przypadku, że $H_0 : R^2 = 0$ (w tym wypadku R^2 odnosi się do tego współczynnika w populacji). Odpowiednią statystykę testową obliczyć możemy według wzoru:

$$F = \frac{(N - p - 1)R^2}{p(1 - R^2)}$$

I (przy założeniu H_0) ma ona rozkład F o p i $N - p - 1$ stopniach swobody. Warto tutaj zauważyć, że statystyczna istotność testu zależy tu nie wyłącznie od uzyskanego w próbie R^2 oraz liczebności próby, ale także od liczby predyktorów! Im mamy ich więcej, tym trudniej uzyskać nam statystycznie istotny wynik.

Zobaczmy, czy uda nam się uzyskać taki sam wynik jak wyliczył nam R.

```
fstat <- ((nrow(data) - 3 - 1) * 0.8865) / (3 * (1 - 0.8865))
fstat
```

```
## [1] 119.7621
```

I teraz sprawdźmy istotność statystyczną dla obliczonej statystyki testowej F :

```
pf(fstat, 3, nrow(data) - 3 - 1, lower.tail = F) * 2
```

```
## [1] 1.910852e-21
```

Uzyskaliśmy wynik mniejszy niż 0.05, co daje nam podstawy do odrzucenia hipotezy zerowej.

Wielkość próbki

Jak dużą powinniśmy mieć próbę, kiedy w naszym badaniu chcemy zastosować regresję liniową? Odpowiedź na to pytanie nie jest jednoznaczna, ale w literaturze proponuje się kilka „reguł kciuka'', które mogą ułatwić podjęcie decyzji o planowanej liczebności próby.

- co najmniej 10 obserwacji dla każdego predyktora ($N \geq 10p$)
- liczba obserwacji powinna przekraczać liczbę predyktorów o przynajmniej 50 ($N \geq p + 50$)

Możemy również zastosować inne podejście i rozważyć to od strony mocy statystycznej.

(Howell odwołuje się w podręczniku do książki Cohena, Cohen Westa i Aikena, ale nie byłem w stanie znaleźć podanych przez Howella liczb i mam podejrzenie, że je zmyślił. Znalazłem jednak inne przykłady w tej książce, które umiem zreplikować w R. Jeżeli ktoś wie skąd Howell wziął te liczby, to będzie zwolniony z jednej pracy domowej)

Jak dużej potrzebujemy próby, aby mieć 80% szans na wykryce prawdziwego R^2 wynoszącego 0.2 przy trzech predyktorach? Odpowiedź na to pytanie da nam analiza mocy statystycznej. W R możemy to zrobić za

pomocą dostępnej w pakiecie `pwr` funkcji `pwr.f2.test`. Funkcja ta przyjmuje szereg argumentów, ale trzy z nich mogą być na pierwszy rzut oka mylące.

- `u` - liczba stopni swobody w liczniku, to jest liczba predyktorów, które planujemy
- `v`- liczba stopni swobody w liczniku, to jest nasze $N - p - 1$ (proszę spojrzeć na wzór na statystykę testową F)
- `f2` - to jest wielkość efektu, którą oblicza się według wzoru $\frac{R^2}{1-R^2}$ (znów, prosze spojrzeć na wzór)

Teraz, kiedy już wiemy co oznaczają poszczególne argumenty tej funkcji, możemy z niej skorzystać. Musimy dokładnie jeden argument tej funkcji pozostawić pustym, my chcemy się dowiedzieć czegoś o wielkości próby, więc pustym pozostawiamy `v`.

```
library(pwr)
pwr.f2.test(u = 3, v = NULL, f2 = 0.2/ (1 -0.2), power = 0.8, sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 3
##               v = 43.70444
##              f2 = 0.25
##       sig.level = 0.05
##           power = 0.8
```

Otrzymaliśmy `v` wynoszące 44 (po zaokrągleniu do góry) co pozwala nam łatwo obliczyć niezbędną liczebność próby: $44 + 3 + 1 = 48$.

W jaki sposób liczba predyktorów w naszym modelu wpływa na moc testu? Rozważmy dwa identyczne przypadki. Interesuje nas wykryce z prawdopodobieństwiem 0.8 „prawdziwego'' R^2 wynoszącego przynajmniej 0.09. Ile obserwacji potrzebujemy dla jednego predyktora?

```
pwr.f2.test(u = 1, v = NULL, f2 = (0.3)**2/ (1 -(0.3**2)), power = 0.8, sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 1
##               v = 79.3268
##              f2 = 0.0989011
##       sig.level = 0.05
##           power = 0.8
```

$80+1+1$ czyli 82. Czy liczebność wzrośnie, kiedy założymy, że będziemy mieć nie jeden ale pięć predyktorów?

```
pwr.f2.test(u = 5, v = NULL, f2 = (0.3)**2/ (1 -(0.3**2)), power = 0.8, sig.level = 0.05)
```

```
##
##      Multiple regression power calculation
##
##               u = 5
##               v = 129.3443
##              f2 = 0.0989011
##       sig.level = 0.05
##           power = 0.8
```

Jak widzimy teraz potrzebujemy $130 + 5 + 1$ czyli 136 obserwacji. Widzimy więc, że im więcej predyktorów, tym większej próby potrzebujemy, żeby zachować taką samą moc. W przypadku regresji najlepszą „regułą kciuka'' jest jednak „im więcej, tym lepiej'' (ale też bez przesady!).

Cząstkowa i półcząstkowa korelacja

Korelacja cząstkowa to korelacja między dwiema zmiennymi (X i Y) przy usuniętym wpływie jednej lub większej liczby zmiennych (Z). Jest to współczynnik korelacji Pearsona pomiędzy składnikami resztowymi (resztami) X_{res} i Y_{res} z modeli regresji liniowej prognozujących odpowiednio X za pomocą Z oraz Y za pomocą Z.

Proszę zwrócić uwagę, że robiliśmy już to (obliczaliśmy korelację między resztami) w poprzednim notatniku, kiedy zajmowaliśmy się podstawami wielokrotnej regresji, dlatego nie będę powtarzał tutaj kodu i odsyłam do poprzedniego notatnika. Korelacja cząstkowa jest o tyle przydatna, że pozwala nam np. stwierdzić, czy, „oczyszczając'' nasze obserwacje z wpływu jakiejś trzeciej zmiennej, zachodzi relacja między dwiema zmiennymi.

Na początek przypomnijmy sobie jak wyglądał model z dwiema zmiennymi:

```
pander(summary(lm(SATcombined ~ Expend + LogPctSAT, data = data)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1147	16.7	68.68	8.447e-49
Expend	11.13	3.264	3.409	0.001346
LogPctSAT	-78.2	4.471	-17.49	3.292e-22

Table 15: Fitting linear model: SATcombined ~ Expend + LogPctSAT

Observations	Residual Std. Error	R^2	Adjusted R^2
50	25.78	0.8861	0.8813

W R możemy łatwo obliczyć korelację cząstkową (i od razu przeprowadzić test!) za pomocą funkcji `pcor.test` z pakietu `ppcor` (w kolumnie `estimate` znajdziemy współczynnik korelacji):

```
library(ppcor)
pcor.test(data$SATcombined, data$Expend, data$LogPctSAT)
```

```
##    estimate    p.value statistic  n gp  Method
## 1 0.4452659 0.00134635  3.409197 50  1 pearson
```

Widzimy, że korelacja między `SATcombined` a `Expand` jest statystycznie istotna przy usuniętym wpływie `LogPctSAT`. Możemy to również obliczyć w drugą stronę:

```
pcor.test(data$SATcombined, data$LogPctSAT, data$Expend)
```

```
##     estimate      p.value statistic  n gp  Method
## 1 -0.9310325 3.292077e-22 -17.49027 50  1 pearson
```

Również uzyskaliśmy statystycznie istotny wynik, ale współczynnik korelacji jest znacznie wyższy.

Innym typem korelacji jest korelacja semicząstkowa. Największą różnicą między korelacją cząstkową a semicząstkową jest to, że w przypadku tej pierwszej usuwamy wpływ trzeciej zmiennej zarówno z predyktora jak i zmiennej objaśnianej, podczas gry w przypadku tej drugiej jedynie z predyktora.

Jeżeli więc obliczymy korelację semiczęstkową między `SATcombined` i `Expend` kontrolując `LogPctSAT`, to jest to korelacja między `SATcombined` a tą częścią `Expend`, która jest niezależna od `LogPctSAT`.

```
spcor.test(data$SATcombined, data$Expend, data$LogPctSAT)
```

```
##    estimate   p.value statistic  n gp  Method
## 1 0.1678232 0.2490611  1.167091 50  1 pearson
```

W jaki sposób interpretować wielkość tych współczynników? W przypadku cząstkowej korelacji, jeżeli podniesiemy do kwadratu współczynnik r, to uzyskamy:

```
pcor.test(data$SATcombined, data$Expend, data$LogPctSAT)$estimate **2
```

```
## [1] 0.1982617
```

Oznacza to, że 20% zmienności w wynikach SAT, której nie da się wyjaśnić za pomocą logarytmu z odsetka osób podchodzacych do testów w danym stanie (`LogPctSAT`), może być wyjaśniona przez tę część nakładów finansowych na edukację (`Expend`), której nie da się wyjaśnić za pomocą logarytmu z odsetka osó" podchodzacych do testów w danym stanie.

Warto zwrócić uwagę, że w kontekście modelu liniowego, test istotności statystycznej dla korelacji cząstkowej dają taki sam wynik, jak testy dla współczynników regresji. Jest to nieprzypadkowe, jeżeli przypomnimy sobie poprzedni notatnik w którym właśnie używając procedury obliczania korelacji cząstkowej tłumaczyliśmy sobie jak działa wielokrotna regresja!

Porównywanie modeli

Jednym z najczęściej przydarzających się problemów związanych z analizą regresji jest to, który z alternatywnych modeli powinniśmy wybrać. Możemy na przykład mieć dwie teorie które mają odmienne predykcje dotyczące tego, które zmienne będą odpowiadać za zmienną objaśnianą. Możemy oczywiście po prostu porównywać R^2 obu modelu, ale wiemy, że co do zasady zwiększenie liczby predyktorów zwiększy również dopasowanie modelu.

Rozważmy ten problem w dwóch sytuacjach. Pierwsza z nich to sytuacja, w której mamy dwa modele, ale zbiór predyktorów jednego z nich zawiera się w zbiorze predyktorów drugie. Taką sytuację nazywać będziemy modelami zagnieżdżonymi. Druga możliwosc jest taka, że zbiór predyktorów jednego nie zawiera się w zbiorze predyktorów drugiego.

Modele zagnieżdżone (hierarchiczne)

W tym wypadku sytuacja jest relatywnie prosta. Rozważmy dwa modele. W pierwszym z nich w zbiorze predyktorów uwzględnimy zmienne `Salary`, `PTratio`, `LogPctSAT` oraz `Expend`. W drugim zaś będziemy mieli tylko `LogPctSAT` oraz `Expend`. Jak widzimy zbiór predyktorów drugiego modelu zawiera się w zbiorze predyktorów pierwszego, więc mamy do czynienia z zagnieżdżonymi modelami.

Pierwszy z nich wygląda tak:

```
fit1 <- lm(SATcombined ~ Salary + PTratio + LogPctSAT + Expend, data = data)
pander(summary(fit1))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1144	42.39	26.99	1.956e-29
Salary	1.622	1.893	0.8569	0.396
PTratio	-0.7883	2.523	-0.3125	0.7561
LogPctSAT	-79.77	4.823	-16.54	9.521e-21
Expend	5.133	8.406	0.6106	0.5446

Table 17: Fitting linear model: SATcombined ~ Salary + PTratio + LogPctSAT + Expend

Observations	Residual Std. Error	R^2	Adjusted R^2
50	26.09	0.8884	0.8784

Drugi zaś tak:

```
fit2 <- lm(SATcombined ~ LogPctSAT + Expend, data = data)
pander(summary(fit2))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1147	16.7	68.68	8.447e-49
LogPctSAT	-78.2	4.471	-17.49	3.292e-22
Expend	11.13	3.264	3.409	0.001346

Table 19: Fitting linear model: SATcombined ~ LogPctSAT + Expend

Observations	Residual Std. Error	R^2	Adjusted R^2
50	25.78	0.8861	0.8813

Widzimy, że pierwszy z nich ma wyższy współczynnik R^2. Czy jednak różnica ta jest statystycznie istotna? Pominiemy tutaj (dośc wyjątkowo jak na dzisiejszy notebook) przeprowadzanie odpowiedniego testu na piechote i skorzystamy z funkcji `anova` pochodzacej z biblioteki standardowej. Jako argumenty przyjmuje ona jeden lub więcej obiektów z modelami.

```
anova(fit1, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: SATcombined ~ Salary + PTratio + LogPctSAT + Expend
## Model 2: SATcombined ~ LogPctSAT + Expend
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     45 30623
## 2     47 31242 -2   -618.72 0.4546 0.6376
```

Widzimy, że otrzymana przez nas wartość p nie przekroczyła progu statystycznej istotności. Oznacza to, że w tym wypadku dodanie dodatkowych predyktorów nie zwiększyło istotnie dopasowania naszego modelu.

Kryterium informacyjne Akaikego (AIC, Aikake's Information Criterion)

Co jednak zrobić, kiedy nie mamy do czynienia z modelami zagnieżdżonymi? Tutaj z pomocą przychodzą inne techniki porównywania ze sobą modeli. Jednym z narzędzi, którego możemy użyć jest kryterium informacyjne Akaikego. Rozważmy sytuację, w której mamy dwa modele - różniące się tym, czy użyliśmy zmiennej `LogPctSAT` czy `PctSAT` (logarytm z odsetka czy sam odsetek).

Możemy obliczyć wartość AIC za pomocą funkcji AIC.

```
fit1 <- lm(SATcombined ~ LogPctSAT + Expend, data = data)
fit2 <- lm(SATcombined ~ PctSAT + Expend, data = data)

pander(AIC(model1, model2))
```

Quitting from lines 296-300 (20_Wielokrotna_regresja_liniowa_cz2.Rmd) Błąd w poleceniu 'AIC(model1, model2)':nie znaleziono obiektu 'model1' Wywołania: ... eval_with_user_handlers -> eval -> eval -> pander -> AIC

Im mniejsza wartość AIC tym lepiej. Widzimy, że w naszym wypadku wersja z `LogPctSAT` ma mniejszy AIC niż wersja z `PctSAT`, co daje nam powody sądzić, że to właśnie ten model jest lepiej dopasowany.

(Jeżeli ktoś odkryje dlaczego SPSS Howella zwraca inną wartość niż R, to również ma odpuszczoną pracę domową).