

Wielokrotna (wieloraka) regresja liniowa

Na dzisiejszych zajęciach opowiemy sobie trochę o wielokrotnej (wielorakiej) regresji liniowej (ang. *multiple linear regression*). Pracować będziemy na przykładzie z rozdziału 15 podręcznika Howella, który znajduje się na stronie w materiałach dostępnych do pobrania po zalogowaniu. Proszę Państwa o przeczytanie tego rozdziału z podręcznika. Kod, który wklejam poniżej, ma stanowić bazę do ćwiczeń i ilustrację, jak różne rzeczy omówione w książce możemy zrobić w R. Kod będzie objaśniony, ale nie zastąpi (!) to lektury podręcznika. Dodatkowo pomyślałem, że możnaby przemycić trochę ciekawych i przydatnych R-owych bibliotek oraz technik, więc będę czasami korzystać z nieznanych Państwu funkcji, ale proszę się nie martwić – wszystkie będę objaśniał i tłumaczył co robią.

Na początek wczytajmy dwie biblioteki, których użyjemy do stworzenia tego pliku oraz dane znajdujące się w pliku `Tab15-1.dat`.

Biblioteka **pander** służy do ładnego wyświetlania wyników różnych testów statystycznych. Bez niej *output* różnych operacji, które wykonujemy, wygląda jak wydruk kodu źródłowego programu (szarawe tło oraz czcionka o stałej szerokości). Pander sprawia, że zamiast wydruku testy statystyczne zwracają ładnie sformatowane wyniki w formie np. tabelki. Z pakietu **knitr** będziemy używać funkcji **kable**, która w zasadzie służy do tego samego, czyli do ładnego wyświetlania danych w dokumentach HTML (takich jak ten). Od razu mogą Państwo zobaczyć działanie tego pakietu – dzięki funkcji **kable** nasza ramka danych jest estetycznie sformatowana.

Uwaga! Wczytaliśmy dane za pomocą funkcji `read.table`. Funkcja ta pozwala wczytać pliki, w których wartości oddzielone są białymi znakami (np. tabulatorami). Domyślnie jednak wczytuje pliki bez nagłówka (zakłada, że wszystkie wiersze to dane). My jednak ustawiliśmy argument `header` na `TRUE`, co sprawiło, że R potraktował pierwszy wiersz jako wiersz z nazwami kolumn.

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

id	State	Expend	P	Ratio	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	
7	Conn	8.817	14.4	50.045	81	431	477	908	3	21.7	4.394	
8	Del	7.030	16.6	39.076	68	429	468	897	3	21.0	4.220	
9	Florida	5.718	19.1	32.588	48	420	469	889	36	20.7	3.871	
10	Georgia	5.193	16.3	32.291	65	406	448	854	16	20.2	4.174	
11	Hawaii	6.078	17.9	38.518	57	407	482	889	17	21.6	4.043	
12	Idaho	4.210	19.1	29.783	15	468	511	979	62	21.4	2.708	
13	Illino	6.136	17.3	39.431	13	488	560	1048	69	21.2	2.565	
14	Indiana	5.826	17.5	36.785	58	415	467	882	19	21.2	4.060	
15	Iowa	5.483	15.8	31.511	5	516	583	1099	64	22.1	1.609	
16	Kansas	5.817	15.1	34.652	9	503	557	1060	74	21.7	2.197	
17	Ken	5.217	17.0	32.257	11	477	522	999	65	20.1	2.398	

id	State	Expend	PTratio	Salary	PctSAT	Verbal	Math	SATcombined	PctACT	ACTcombined	LogPctSAT
18	Louisia	4.761	16.8	26.461	9	486	535	1021	80	19.4	2.197
19	Maine	6.428	13.8	31.972	68	427	469	896	2	21.5	4.220
20	Marylan	7.245	17.0	40.661	64	430	479	909	11	20.7	4.159
21	Mass	7.287	14.8	40.795	80	430	477	907	6	21.6	4.382
22	Mich	6.994	20.1	41.895	11	484	549	1033	68	21.3	2.398
23	Minn	6.000	17.5	35.948	9	506	579	1085	60	22.1	2.197
24	Mississ	4.080	17.5	26.818	4	496	540	1036	79	18.7	1.386
25	Missour	5.383	15.5	31.189	9	495	550	1045	64	21.5	2.197
26	Montana	5.692	16.3	28.785	21	473	536	1009	55	21.9	3.045
27	Neb	5.935	14.5	30.922	9	494	556	1050	73	21.7	2.197
28	Nev	5.160	18.7	34.836	30	434	483	917	39	21.3	3.401
29	NH	5.859	15.6	34.720	70	444	491	935	4	22.3	4.248
30	NJ	9.774	13.8	46.087	70	420	478	898	3	20.8	4.248
31	NM	4.586	17.2	28.493	11	485	530	1015	59	20.3	2.398
32	NY	9.623	15.2	47.612	74	419	473	892	16	21.9	4.304
33	NC	5.077	16.2	30.793	60	411	454	865	11	19.3	4.094
34	ND	4.775	15.3	26.327	5	515	592	1107	78	21.4	1.609
35	Ohio	6.162	16.6	36.802	23	460	515	975	60	21.3	3.135
36	Ok	4.845	15.5	28.172	9	491	536	1027	66	20.6	2.197
37	Oregon	6.436	19.9	38.555	51	448	499	947	12	22.3	3.932
38	Penn	7.109	17.1	44.510	70	419	461	880	8	21.0	4.248
39	RI	7.469	14.7	40.729	70	425	463	888	2	21.4	4.248
40	SC	4.797	16.4	30.279	58	401	443	844	13	18.9	4.060
41	SD	4.775	14.4	25.994	5	505	563	1068	68	21.3	1.609
42	Tenn	4.388	18.6	32.477	12	497	543	1040	83	19.7	2.485
43	Texas	5.222	15.7	31.223	47	419	474	893	30	20.2	3.850
44	Utah	3.656	24.3	29.082	4	513	563	1076	69	21.5	1.386
45	Vermont	6.750	13.8	35.406	68	429	472	901	7	21.9	4.220
46	Va	5.327	14.6	33.987	65	428	468	896	6	20.7	4.174
47	Wash	5.906	20.2	36.151	48	443	494	937	16	22.4	3.871
48	WV	6.107	14.8	31.944	17	448	484	932	57	20.0	2.833
49	Wisc	6.930	15.9	37.746	9	501	572	1073	64	22.3	2.197
50	Wyoming	6.160	14.9	31.285	10	476	525	1001	70	21.4	2.303

Wczytane przez nas dane dotyczą edukacji w Stanach Zjednoczonych i zostały zebrane przez Gubner (1999). Jak widzimy, w każdym wierszu mamy informacje dotyczące jednego stanu.

Nas dzisiaj będą interesowały dane z kolumn `SATcombined`, `PTratio`, `PctSAT`, oraz `LogPctSAT`.

W kolumnie `SATcombined` znajduje się średni wynik testu SAT w danym stanie. Test ten umożliwia zarekrutowanie się na studia wyższe. Nie jest to jednak jedyny taki test. Uczniowie mogą również zdawać test ACT. Generalnie rzecz biorąc test SAT jest popularniejszy w jednych stanach, w innych zaś większą popularnością cieszy się ACT. Różnice w popularności możemy ocenić przyglądając się wartościom z kolumny `PctSAT`. Znajduje się tam odsetek uczniów, którzy przystąpili do egzaminu SAT. W kolumnie `LogPctSAT` znajdziemy logarytm naturalny z tej wartości. Dodatkowo warto pamiętać, że SAT wymagany jest przez wiele prestiżowych uczelni. W kolumnie `PTratio` znajdziemy informacje o tym ilu uczniów przypada na jednego nauczyciela. W kolumnie `Expend` znajdują się wydatki na edukację w danym stanie.

Zanim przystąpimy do analizy danych, obejrzymy rozkład naszych zmiennych. Dla każdej z nich narysujemy histogram, wykres kwantyl-quantyl (który posłuży nam do oceny normalności rozkładu) oraz wykres rozrzutu, gdzie zmienną na osi Y będzie `SATcombined` (oprócz samego `SATcombined`).

Żeby to zrobić, posłużymy się odpowiednimi funkcjami z pakietu `ggplot2`. Jeżeli nigdy nie korzystali Państwo z tego pakietu, to zachęcam do przerobienia odpowiedniego notebooka ze strony kursu (nagrałem też do nich

filmiki, które dostępne są na YT). Dla tych z Państwa, którzy korzystali z tego pakietu, kilka ciekawych twistów:

- Użyliśmy pakietu `gridExtra` i zawartej w nim funkcji `grid.arrange`, żeby narysować kilka wykresów stworzonych za pomocą `ggplot2` na jednym panelu. Jeśli rysowalibyśmy te wykresy używając standardowych funkcji R takich jak `hist`, to moglibyśmy po prostu użyć `par(mfrow=c(4,3))`. W przypadku `ggplot2` technika ta jednak nie działa.
- Każdy z wykresów jest obiektem, który można przypisać do zmiennej, co zrobiliśmy.
- Wszystkie te obiekty przekazaliśmy do funkcji `grid.arrange`, która narysowała te wykresy.

To, w jaki sposób rysuje się zawarte poniżej typy wykresów w `ggplot2`, omówione jest w notatniku poświęconym temu pakietowi, więc odsyłam do niego (oraz do dokumentacji `ggplot2`!), jeżeli ktoś chciałby dowiedzieć się więcej.

```
library(ggplot2)
library(gridExtra)
expend_qq <- ggplot(data, aes(sample=Expend))+stat_qq() + stat_qq_line()
expend_hist <- ggplot(data, aes(x=Expend))+ geom_histogram(bins = 8)
expend_scatter <- ggplot(data, aes(x=Expend, y=SATcombined)) +geom_point() + geom_smooth(method=lm)

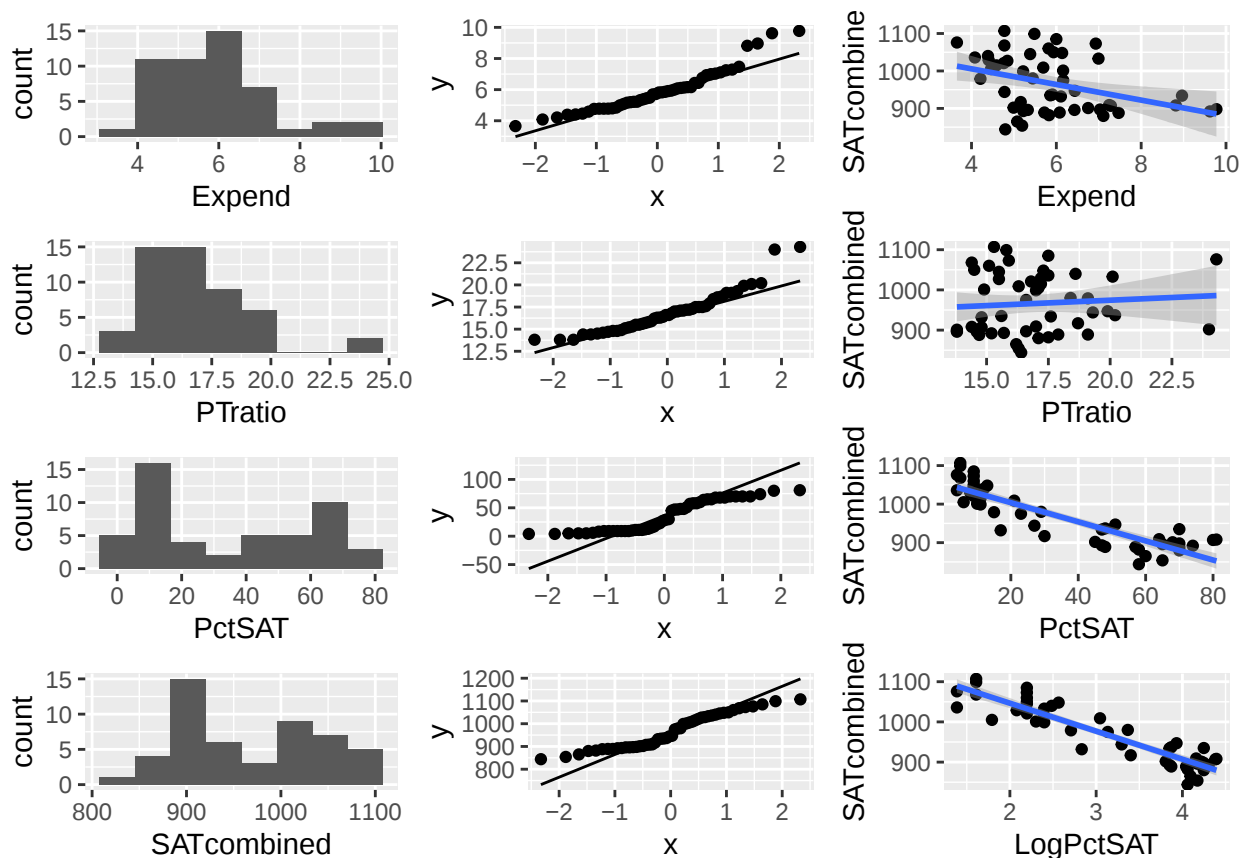
ptratio_qq <- ggplot(data, aes(sample=PTratio))+stat_qq() + stat_qq_line()
ptratio_hist <- ggplot(data, aes(x=PTratio))+ geom_histogram(bins = 8)
ptratio_scatter <- ggplot(data, aes(x=PTratio, y=SATcombined)) +geom_point() + geom_smooth(method=lm)

pctsat_qq <- ggplot(data, aes(sample=PctSAT))+stat_qq() + stat_qq_line()
pctsat_hist <- ggplot(data, aes(x=PctSAT))+ geom_histogram(bins = 8)
pctsat_scatter <- ggplot(data, aes(x=PctSAT, y=SATcombined)) +geom_point() + geom_smooth(method=lm)

satcombined_qq <- ggplot(data, aes(sample=SATcombined))+stat_qq() + stat_qq_line()
satcombined_hist <- ggplot(data, aes(x=SATcombined))+ geom_histogram(bins = 8)
logpctsat_scatter <- ggplot(data, aes(x=LogPctSAT, y=SATcombined)) +geom_point() + geom_smooth(method=lm)

grid.arrange(expend_hist, expend_qq, expend_scatter, # dajemy tutaj wszystkie wykresy
              ptratio_hist, ptratio_qq, ptratio_scatter,
              pctsat_hist, pctsat_qq, pctsat_scatter,
              satcombined_hist, satcombined_qq, logpctsat_scatter,
              ncol = 3 # liczba kolumn, w które chcemy je ułożyć
              )

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



Rzeczą, która powinna zwrócić naszą uwagę jest to, że relacja między `Expend` i `SATcombined` jest negatywna, to znaczy im więcej pieniędzy wydajemy na edukację, tym gorsze wyniki uzyskują uczniowie (sic!). Jest to dość kontrintuicyjne, ponieważ spodziewalibyśmy się zupełnie odwrotnego efektu – wydatki na edukację powinny przecież polepszać jej jakość.

Spróbujemy teraz obliczyć współczynnik korelacji, między zmiennymi, które nas interesują. Jak Państwo wiedzą, do obliczenia współczynnika korelacji oraz przeprowadzania odpowiedniego testu statystycznego ($H_0 : \rho = 0$) używamy funkcji `cor.test`. Dla przykładu możemy obliczyć korelację między zmienną `SATcombined` oraz `Expend`:

```
pander(cor.test(~ SATcombined + Expend, data = data))
```

Table 2: Pearson's product-moment correlation: `SATcombined` and `Expend`

Test statistic	df	P value	Alternative hypothesis	cor
-2.851	48	0.006408 * *	two.sided	-0.3805

Widzimy, że uzyskaliśmy negatywny współczynnik korelacji — niezbyt wysoki, ale statystycznie istotny! Jest to dość zaskakujące.

Efekt ten, przynajmniej według podręcznika, z którego pochodzi ten przykład, związany jest z faktem, że w tych stanach, w których mały odsetek uczniów przystępuje do testu SAT, przystępują do niego zazwyczaj najlepsi i to zaburza wynik. W dalszej części notatnika pokażemy w jaki sposób możemy za pomocą wielokrotnej regresji to pokazać!

Krótką uwagę na marginesie. Gdybyśmy mieli policzyć współczynnik korelacji dla wszystkich kombinacji zmiennych, musielibyśmy napisać bardzo dużo kodu! Możemy sobie z tym jednak poradzić. W pakiecie

`rstatix` znajduje się sporo ciekawych funkcji do „masowego” przeprowadzania testów statystycznych. My skorzystamy z funkcji `cor_test` (która jest odpowiednikiem `cor.test` z biblioteki standardowej R).

```
library(rstatix)
# Do funkcji cor_test przekazujemy ramkę danych oraz nazwy kolumn.
results <- cor_test(data = data, Expend, PTratio, Salary, PctSAT, SATcombined, LogPctSAT)
# Dane wyjściowe `cor_test` to ramka danych więc możemy ją sformatować za pomocą `kable`
kable(results[, c("var1", "var2", "cor", "p")])
```

var1	var2	cor	p
Expend	Expend	1.0000	0.00e+00
Expend	PTratio	-0.3700	7.99e-03
Expend	Salary	0.8700	0.00e+00
Expend	PctSAT	0.5900	5.80e-06
Expend	SATcombined	-0.3800	6.41e-03
Expend	LogPctSAT	0.5600	2.27e-05
PTratio	Expend	-0.3700	7.99e-03
PTratio	PTratio	1.0000	0.00e+00
PTratio	Salary	-0.0011	9.94e-01
PTratio	PctSAT	-0.2100	1.37e-01
PTratio	SATcombined	0.0810	5.75e-01
PTratio	LogPctSAT	-0.1300	3.61e-01
Salary	Expend	0.8700	0.00e+00
Salary	PTratio	-0.0011	9.94e-01
Salary	Salary	1.0000	0.00e+00
Salary	PctSAT	0.6200	1.90e-06
Salary	SATcombined	-0.4400	1.39e-03
Salary	LogPctSAT	0.6100	2.20e-06
PctSAT	Expend	0.5900	5.80e-06
PctSAT	PTratio	-0.2100	1.37e-01
PctSAT	Salary	0.6200	1.90e-06
PctSAT	PctSAT	1.0000	0.00e+00
PctSAT	SATcombined	-0.8900	0.00e+00
PctSAT	LogPctSAT	0.9600	0.00e+00
SATcombined	Expend	-0.3800	6.41e-03
SATcombined	PTratio	0.0810	5.75e-01
SATcombined	Salary	-0.4400	1.39e-03
SATcombined	PctSAT	-0.8900	0.00e+00
SATcombined	SATcombined	1.0000	0.00e+00
SATcombined	LogPctSAT	-0.9300	0.00e+00
LogPctSAT	Expend	0.5600	2.27e-05
LogPctSAT	PTratio	-0.1300	3.61e-01
LogPctSAT	Salary	0.6100	2.20e-06
LogPctSAT	PctSAT	0.9600	0.00e+00
LogPctSAT	SATcombined	-0.9300	0.00e+00
LogPctSAT	LogPctSAT	1.0000	0.00e+00

Przejdźmy teraz do głównego tematu dzisiejszych zajęć, to znaczy do wielorakiej regresji liniowej. Jak już wspomnieliśmy, chcielibyśmy zobaczyć związek między wydatkami na szkolnictwo i testem SAT kontrolując przy tym odsetek przystępujących do SAT uczniów, to znaczy zmienną `LogPctSAT` (logarytm z odsetka przystępujących osób, wybraliśmy go bo ma troszkę lepszy rozkład i zależność jest bardziej liniowa).

Stworzymy dwa modele w których zmienną objaśnianą jest `SATcombined`. W pierwszym z nich (`fit1`) uwzględnić będziemy tylko `Expend`.

```
fit1 <- lm(SATcombined ~ Expend, data = data)
pander(summary(model1))
```

Quitting from lines 97-99 (19_Wielokrotna_regresja liniowa.Rmd) Błąd w poleceniu 'summary(model1)':nie znaleziono obiektu 'model1' Wywołania: ... eval_with_user_handlers -> eval -> eval -> pande -> summary

Teraz stworzymy drugi model, w którym to uwzględnimy jednak dodatkową zmienną - LogPctSAT.

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

	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 5: Fitting linear model: SATcombined ~ Expend + LogPct-SAT

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

Widzimy, że zmiana jest gigantyczna! W poprzednim wypadku współczynnik regresji był dla **Expand** ujemny (-20.89) ale teraz, kiedy kontrolujemy zmienną **LogPctSAT** jest delikatnie dodatni (11.13). To jest dokładnie to, czego oczekiwaliśmy! Relacja między **LogPctSAT** a **SATcombined** jest negatywna - im mniej osób przystępuje w danym stanie do egzaminu, tym wyższy wynik średnio uzyskują. To również jest zgodne z naszymi oczekiwaniami. Warto zwrócić uwagę, że dramatycznie zwiększyło się również dopasowanie naszego modelu do danych - współczynnik R^2 wzrósł z 0.145 do 0.886.

Interpretacja wielokrotnej regresji

Spróbujmy prześledzić, co tak naprawdę robi wielokrotna regresja liniowa po to, aby zrozumieć jak interpretować jej wyniki.

Stworzymy model, w którym zmienną objaśnianą będzie **SATcombined** a predyktorem **LogPctSAT**.

```
fit_sat <- lm(SATcombined ~ LogPctSAT, data = data)
```

Obliczymy jakie wartości dla **SATcombined** przewiduje nasz model. Posłużymy się tutaj funkcją **predict**. Działa ona tak, jakbyśmy po prostu wzięli wyraz wolny regresji oraz współczynniki kierunkowe a następnie dla każdej obserwacji obliczali przewidywaną wartość.

```
predictsat <- predict(fit_sat)
```

Następnie obliczymy różnicę między prawdziwymi (rzeczywiście zaobserwowanymi) wartościami a wartościami przewidywanymi przez regresję (czyli residua/reszty!). Przypiszmy je do zmiennej **residsat**.

```
residsat <- data$SATcombined - predictsat
```

W tym momencie musimy dokonać ważnej obserwacji. Wartości naszej nowej zmiennej (**residstat**) są niezależne od wartości **LogPctSAT** (ponieważ w pewnym sensie usunęliśmy z nich wpływ **LogPctSAT**). Dokładnie to samo możemy zrobić z naszą drugą zmienną, czyli **Expend**.

```
fit_expend <- lm(Expend ~ LogPctSAT, data = data)
predictexpend <- predict(fit_expend)
residexpend <- data$Expend - predictexpend
```

Mamy teraz dwie zmienne, które są niezależne od `LogPctSat`. W takim razie, możemy stworzyć model, w którym przewidujemy wartości `residsat` na podstawie wartości `residexpend`!

```
fit_resid <- lm(residsat ~ residexpend)
pander(summary(fit_resid))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.526e-13	3.608	-7.001e-14	1
residexpend	11.13	3.23	3.445	0.001194

Table 7: Fitting linear model: `residsat ~ residexpend`

Observations	Residual Std. Error	R^2	Adjusted R^2
50	25.51	0.1983	0.1816

Jak widać współczynnik regresji który uzyskaliśmy dla tego modelu z jedną zmienną jest dokładnie taki sam jak wówczas, gdy mieliśmy dwie zmienne. Dlaczego? Dlatego, że „manualnie” zrobiliśmy to, co regresja robi za nas – obliczyliśmy współczynnik regresji kontrolując efekty, które można przypisać trzeciej zmiennej (bo „usunęliśmy” te efekty zarówno z predyktora jak i ze zmiennej objaśnianej).

O wielokrotnej regresji liniowej możemy też myśleć w inny sposób. Wiemy, że w przypadku naszego modelu z dwoma predyktorami równanie regresji ma postać

$$\hat{Y} = b_1 X_1 + b_2 X_2 + b_0$$

Czyli umując to w kategoriach naszych zmiennych:

$$\widehat{SAT} = b_1 Expend + b_2 LogPctSAT + b_0$$

W naszym wypadku znamy b_1 oraz b_2 (bo są to nasze współczynniki regresji) więc równanie to możemy przedstawić również tak:

$$\widehat{SAT} = 11.129 \times Expend - 78.203 \times LogPctSAT + 1147.100$$

Możemy więc korzystając z tego wzoru obliczyć $\widehat{SAT}$.

```
predictions <- predict(fit2)
```

Obliczmy więc korelację między $\widehat{SAT}$ (naszymi predykcjami w zmiennej `predictions`) oraz `SAT` (czyli wartościami zaobserwowanymi). Tak obliczony współczynnik korelacji podnieśmy do kwadratu.

```
cor.test(data$SATcombined, predictions)$estimate ** 2
```

```
##      cor
## 0.8861068
```

Przypomnijmy sobie jak wyglądał nasz model dla dwóch zmiennych, który stworzyliśmy wcześniej.

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

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1147	16.7	68.68	8.447e-49

	Estimate	Std. Error	t value	Pr(> t)
Expend	11.13	3.264	3.409	0.001346
LogPctSAT	-78.2	4.471	-17.49	3.292e-22

Table 9: Fitting linear model: SATcombined $\sim$ Expend + LogPct-SAT

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

Jak widzimy uzyskaliśmy dokładnie tę samą wartość co w przypadku R^2 wcześniej! Nakierowuje nas to na pewien sposób myślenia o regresji, zgodnie z którym zasadniczo sprowadza się ona do utworzenia nowej zmiennej ($\widehat{SAT}$), będącej najlepszą liniową kombinacją predyktorów w naszym modelu (**Expend** oraz **LogPctSAT**). Przez „najlepszą liniową kombinację” rozumiemy to, że nie ma żadnych innych wag, które moglibyśmy przypisać naszym predyktorom (współczynników regresji), które lepiej przewidywałyby naszą zmienną objaśnianą.

O współczynniku R dla wielokrotnej regresji liniowej możemy myśleć po prostu jako o współczynniku korelacji r Pearsona tyle że dla zmiennej stanowiącej najlepszą liniową kombinację predyktorów.