Analýza a vizualizace ekonomických dat

Michal Kvasnička a Štěpán Mikula

Co se naučíte?

Tento dokument ukazuje několik příkladů toho, co se můžete naučit v předmětu Analýza a vizualizace ekonomických dat. Celý dokument je vytvořen pomocí RMarkdownu (13. přednáška), který umožňuje do jednoho souboru spojit psaný text a analýzu dat. Na zdrojový kód, který byl použit pro vytvoření této stránky se můžete podívat zde.

Data a základní operace s daty

Co je náplní kurzu prakticky ukážeme na datasetu "diamonds", který obsahuje údaje o 53940 diamantech. Data jsou uložena v tabulce, která je uložena v tabulce (`data.frame`) `diamonds` v následujícím formátu:

```
diamonds %>% print(n=5)
```

```
## Source: local data frame [53,940 x 10]
##
##    carat     cut  color clarity depth table price     x     y     z
##    (dbl)  (fctr) (fctr)  (fctr) (dbl) (dbl) (int) (dbl) (dbl) (dbl)
## 1   0.23   Ideal      E     SI2  61.5    55   326  3.95  3.98  2.43
## 2   0.21 Premium      E     SI1  59.8    61   326  3.89  3.84  2.31
## 3   0.23    Good      E     VS1  56.9    65   327  4.05  4.07  2.31
## 4   0.29 Premium      I     VS2  62.4    58   334  4.20  4.23  2.63
## 5   0.31    Good      J     SI2  63.3    58   335  4.34  4.35  2.75
## ..   ...     ...    ...     ...   ...   ...   ...   ...   ...   ...
```

Program R umožňuje s daty lehce manipulovat. Jednoduše si můžete zúžit původní data podle vybraných kritérií. Například se můžeme podívat na cenu a váhu u nejlépe zbarvených kamenů:

```
diamonds %>% filter(color=="D") %>% select(color,price,carat) %>% print(n=5)
```

```
## Source: local data frame [6,775 x 3]
##
##     color price carat
##    (fctr) (int) (dbl)
## 1       D   357  0.23
## 2       D   402  0.23
## 3       D   403  0.26
## 4       D   403  0.26
## 5       D   403  0.26
## ..    ...   ...   ...
```

Data lze i jednoduše řadit. Například je možné se podívat na ty nejdražší kameny:

```
diamonds %>% arrange(desc(price)) %>% print(n=5)
```

```
## Source: local data frame [53,940 x 10]
##
##    carat       cut  color clarity depth table price     x     y     z
##    (dbl)    (fctr) (fctr)  (fctr) (dbl) (dbl) (int) (dbl) (dbl) (dbl)
## 1   2.29   Premium      I     VS2  60.8    60 18823  8.50  8.47  5.16
## 2   2.00 Very Good      G     SI1  63.5    56 18818  7.90  7.97  5.04
## 3   1.51     Ideal      G      IF  61.7    55 18806  7.37  7.41  4.56
## 4   2.07     Ideal      G     SI2  62.5    55 18804  8.20  8.13  5.11
## 5   2.00 Very Good      H     SI1  62.8    57 18803  7.95  8.00  5.01
## ..   ...       ...    ...     ...   ...   ...   ...   ...   ...   ...
```

Jednotlivé operace lze kombinovat do větších celků. Například je možné vybrat nejdražší kámen pro každou barvu:

```
diamonds %>% group_by(color) %>% arrange(desc(price)) %>% slice(1L)
```

```
## Source: local data frame [7 x 10]
## Groups: color [7]
##
##   carat       cut  color clarity depth table price     x     y     z
##   (dbl)    (fctr) (fctr)  (fctr) (dbl) (dbl) (int) (dbl) (dbl) (dbl)
## 1  2.19     Ideal      D     SI2  61.8    57 18693  8.23  8.49  5.17
## 2  2.02 Very Good      E     SI1  59.8    59 18731  8.11  8.20  4.88
## 3  1.71   Premium      F     VS2  62.3    59 18791  7.57  7.53  4.70
## 4  2.00 Very Good      G     SI1  63.5    56 18818  7.90  7.97  5.04
## 5  2.00 Very Good      H     SI1  62.8    57 18803  7.95  8.00  5.01
## 6  2.29   Premium      I     VS2  60.8    60 18823  8.50  8.47  5.16
## 7  3.01   Premium      J     SI2  60.7    59 18710  9.35  9.22  5.64
```

Data lze jednoduše agregovat. Například spočítat průměrnou cenu pro každou barvu kamenů.

```
diamonds %>% group_by(color) %>% summarise(average_price=mean(price,na.rm=TRUE))
```

```
## Source: local data frame [7 x 2]
##
##    color average_price
##   (fctr)         (dbl)
## 1      D      3169.954
## 2      E      3076.752
## 3      F      3724.886
## 4      G      3999.136
## 5      H      4486.669
## 6      I      5091.875
## 7      J      5323.818
```

Kritéria lze i kombinovat. Následující tabulka obsahuje průměrnou cenu kamene pro kombinaci barvy a řezu (pro stručnost je vypsáno jen prvních 10 řádků):

```
diamonds %>% group_by(color,cut) %>% summarise(average_price=mean(price,na.rm=TRUE)) %>% print(n=10)
```

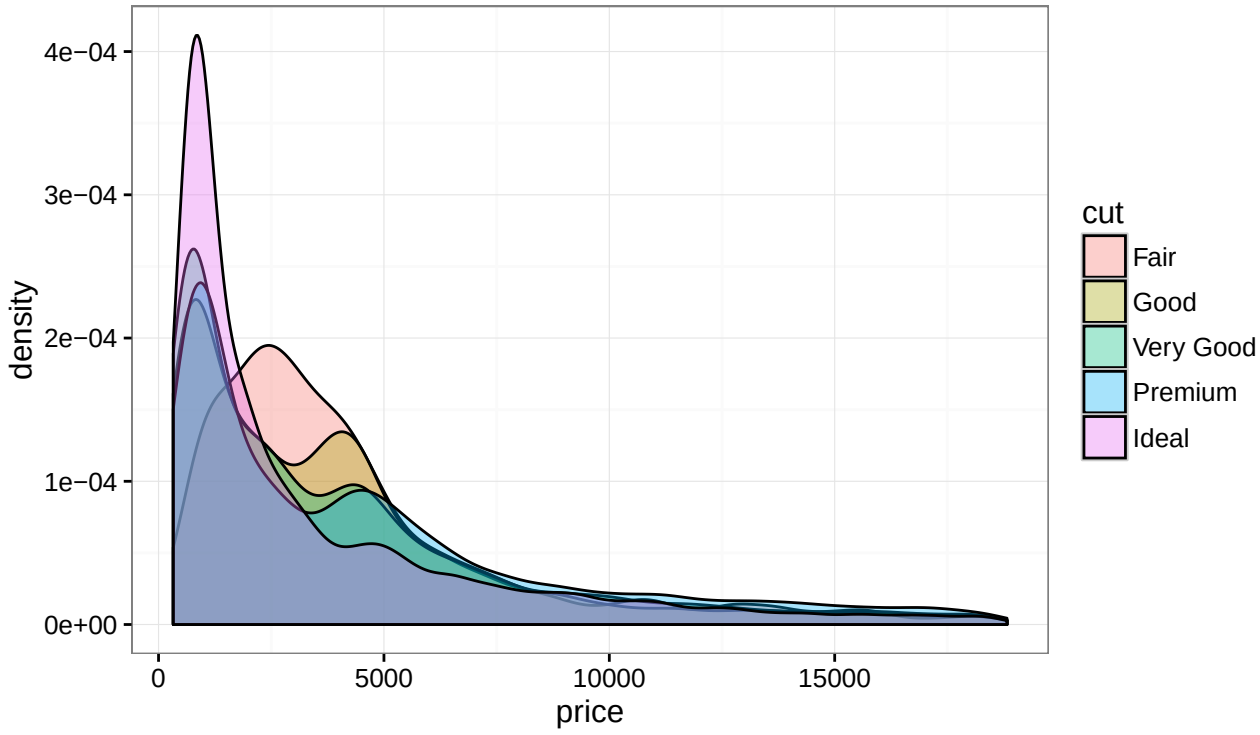
```
## Source: local data frame [35 x 3]
```

```
## Groups: color [?]
##
##     color       cut average_price
##    (fctr)    (fctr)         (dbl)
## 1       D      Fair      4291.061
## 2       D      Good      3405.382
## 3       D Very Good      3470.467
## 4       D   Premium      3631.293
## 5       D     Ideal      2629.095
## 6       E      Fair      3682.312
## 7       E      Good      3423.644
## 8       E Very Good      3214.652
## 9       E   Premium      3538.914
## 10      E     Ideal      2597.550
## ..    ...       ...           ...
```

Vykreslení dat

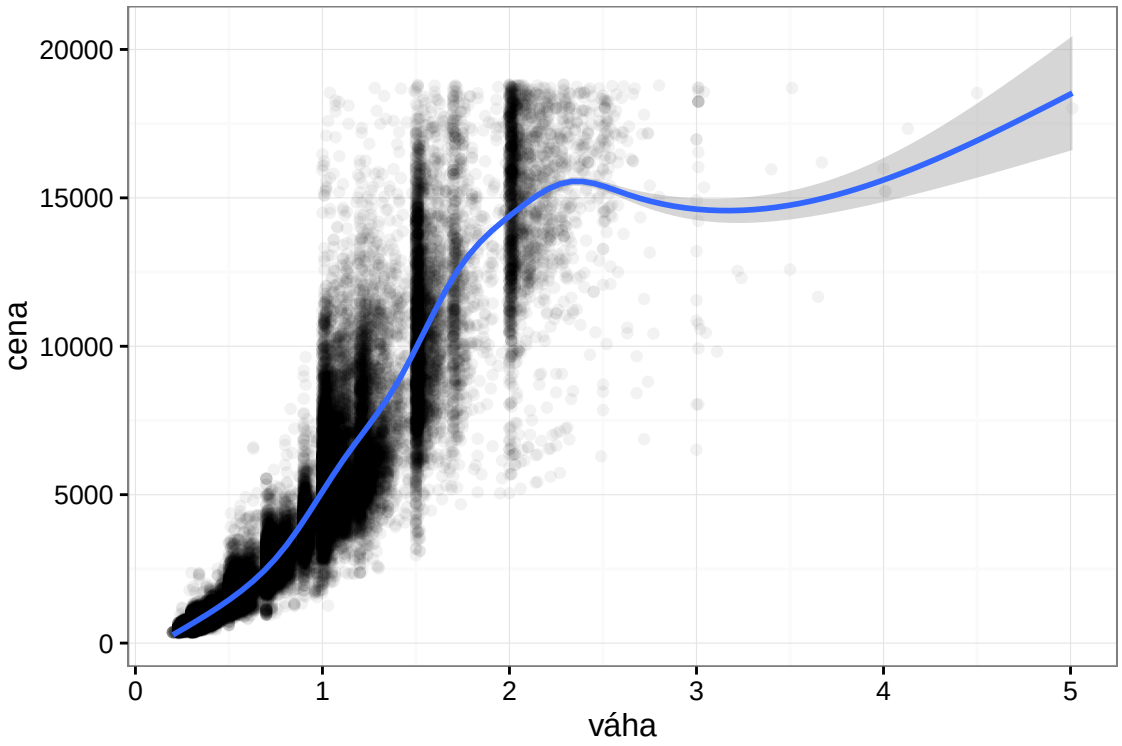
R obsahuje mocné nástroje pro vizualizaci dat a jejich vztahů. Například odhad hustoty rozdělení ceny pro jednotlivé řezy kamenů.

```
diamonds %>%
  ggplot(aes(price,fill=cut)) +
  geom_density(alpha=0.35) +
  theme_bw()
```



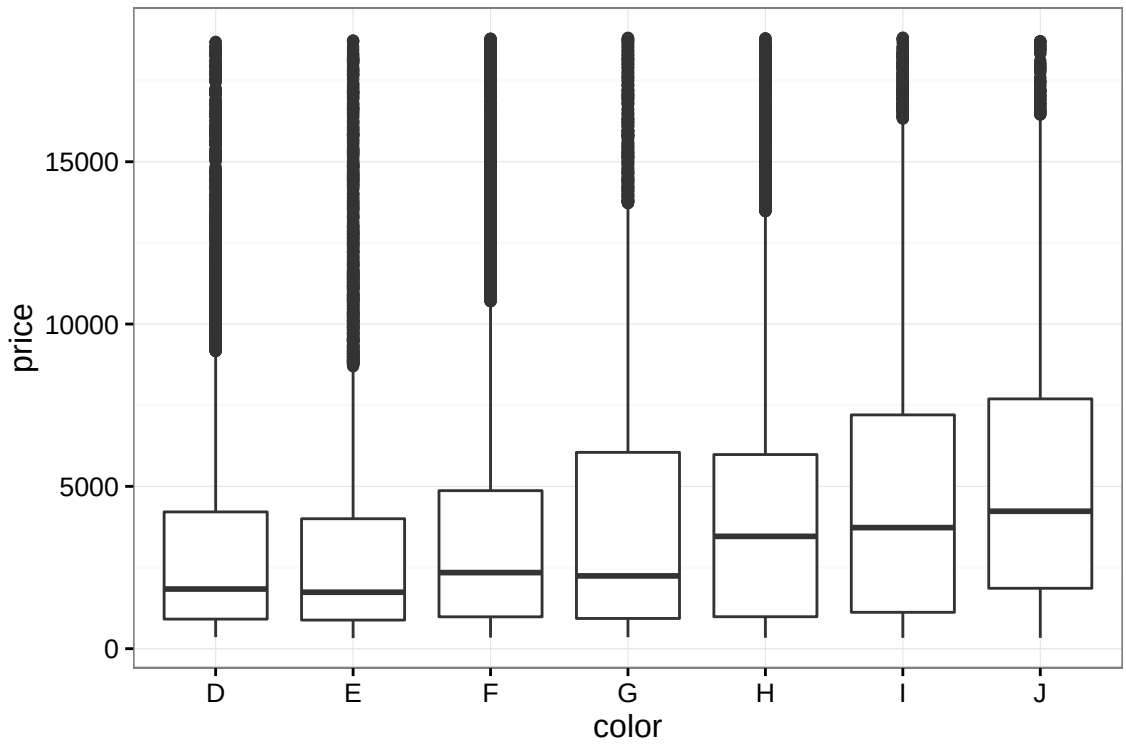
Vzhled grafů lze modifikovat a přidávat dodatečné funkce – v tomto případě vyhlazení. Všimněte si, že R nemá problém s češtinou.

```
diamonds %>%
  ggplot(aes(x=carat,y=price)) +
  geom_point(alpha=0.05) +
  geom_smooth() +
  xlab("váha") + ylab("cena") +
  theme_bw()
```



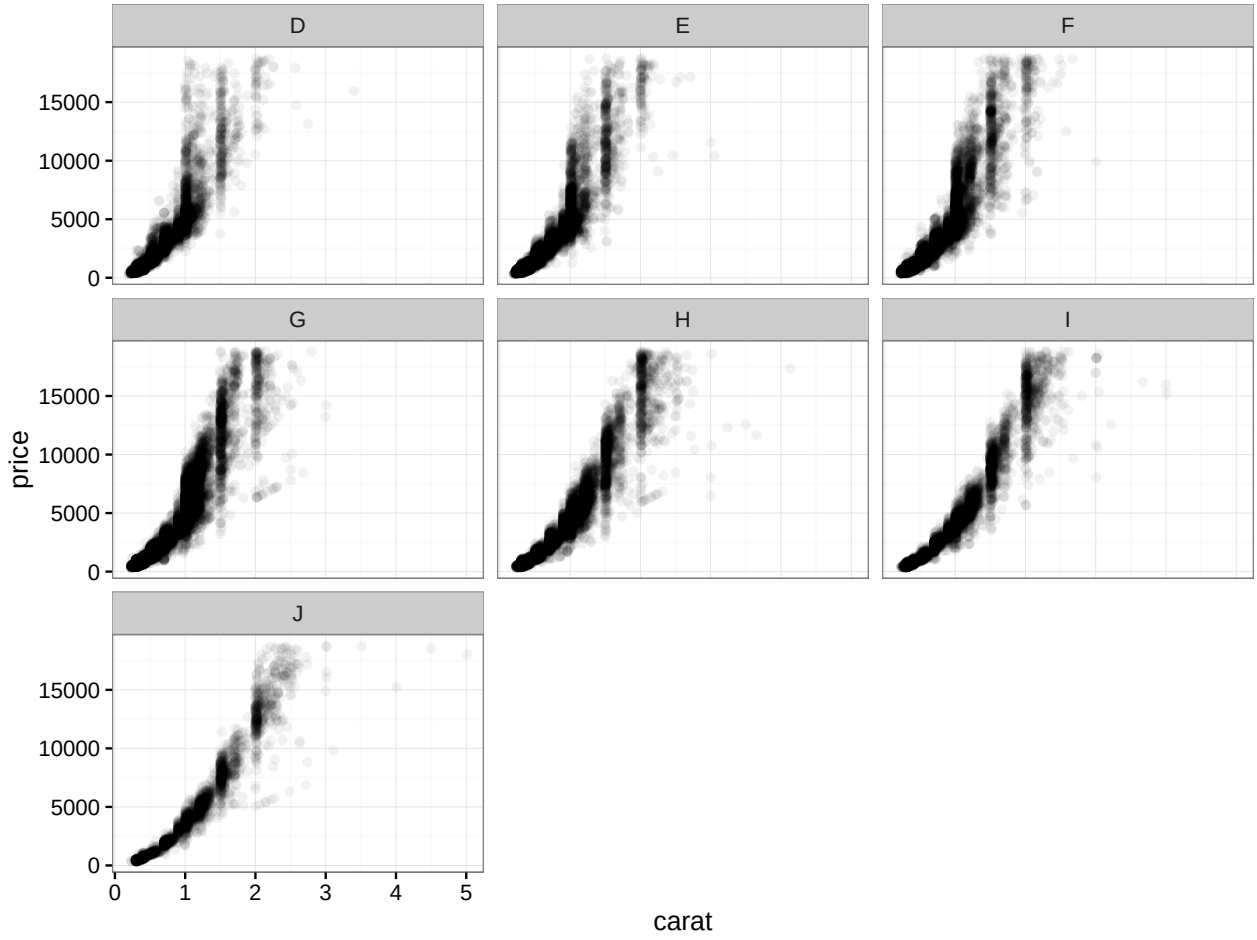
Vykreslovat lze i vztahy kategoriálních proměnných. Například vztah ceny podle barvy kamene ve formě boxplotu...

```
diamonds %>%
  ggplot(aes(color,y=price)) +
  geom_boxplot() +
  theme_bw()
```



... nebo vykreslením zvláštního grafu pro každou barvu.

```
diamonds %>%
  ggplot(aes(x=carat,y=price)) +
  geom_point(alpha=0.05) +
  facet_wrap(~color) +
  theme_bw()
```



Regrese

V diplomových a bakalářských pracích budete často potřebovat provést kvantitativní analýzu dat. S R to není problém. Například ukážeme odhad rovnice vysvětlující cenu diamantů jejich charakteristikami (váhou, řezem, barvou, jasností a velikostí):

$$price = \alpha + \beta carat + \gamma \mathbf{cut} + \epsilon table + \varepsilon$$

```
model <- price ~ carat + cut + table
lm(model, diamonds) -> em
```

Proměnné `cut`, `color` a `clarity` jsou kategoriální. R je automaticky převede na vektor umělých proměnných (a jednu z nich vypustí). Výsledky regrese se dají zobrazit v přehledné tabulce:

Často potřebujete porovnat model odhadnuté v různých specifikacích. Například by mohlo být zajímavé nahradit proměnnou `table` jiným měřítkem velikosti kamene.

```
model <- list(
  model,
  model %>% update(.~.-table+depth),
  model %>% update(.~.-table+x),
  model %>% update(.~.-table+y),
  model %>% update(.~.-table+z)
```

Table 1:

	Dependent variable:
	price
carat	7,878.923***
	(14.046)
cut.L	1,202.788***
	(26.920)
cut.Q	−546.624***
	(23.350)
cut.C	348.862***
	(20.494)
cut^4	58.297***
	(16.495)
table	−19.843***
	(3.551)
Constant	−1,555.609***
	(205.597)
Observations	53,940
R^2	0.857
Adjusted R^2	0.857
Residual Std. Error	1,511.034
F Statistic	53,676.410***

Note:

```
)

lapply(model,function(x) lm(x, data=diamonds)) -> em
```

A výsledky lze opět porovnat v přehledné tabulce...

Table 2:

	Dependent variable:				
	price				
	(1)	(2)	(3)	(4)	(5)
carat	7,878.923***	7,873.249***	10,325.350***	8,983.318***	9,262.455***
	(14.046)	(13.967)	(61.117)	(44.700)	(45.331)
cut.L	1,202.788***	1,148.315***	1,287.265***	1,277.871***	1,164.672***
	(26.920)	(27.518)	(25.724)	(25.977)	(25.957)
cut.Q	−546.624***	−471.615***	−562.039***	−562.390***	−487.173***
	(23.350)	(23.750)	(22.791)	(23.023)	(22.949)
cut.C	348.862***	366.133***	360.965***	388.540***	371.483***
	(20.494)	(20.195)	(19.904)	(20.103)	(20.023)
cut^4	58.297***	87.579***	51.441***	75.033***	90.016***
	(16.495)	(16.271)	(16.000)	(16.138)	(16.093)
table	−19.843***				
	(3.551)				
depth		−50.418***			
		(4.848)			
x			−1,064.184***		
			(25.820)		
y				−483.911***	
				(18.485)	
z					−981.262***
					(30.442)
Constant	−1,555.609***	434.754	1,427.431***	−824.459***	−319.172***
	(205.597)	(301.924)	(101.320)	(73.318)	(75.467)
Observations	53,940	53,940	53,940	53,940	53,940
R^2	0.857	0.857	0.861	0.858	0.859
Adjusted R^2	0.857	0.857	0.861	0.858	0.859
Residual Std. Error	1,511.034	1,509.958	1,488.216	1,501.959	1,497.119
F Statistic	53,676.410***	53,765.760***	55,612.810***	54,435.970***	54,846.710***

Note: