

Predavanja 8 – Analiza podatkov od začetka do konca

Ogledali si bomo primer celotnega procesa obdelave podatkov in osnovno modeliranje podatkov z linearnim modelom. Predavanje bo obravnavalo zajem podatkov, predobdelavo in na koncu modeliranje. Uporabili bomo podatke o oposumih, kjer bomo želeli izvedeti, če lahko s pomočjo informacije o dolžinah oposumovih glav, sklepamo o dolžini njihovih teles.

Branje podatkov in osnovna obdelava

Podatke moramo najprej pridobiti. V našem primeru smo podatke našli na spletni strani kaggle. Podatke bi lahko tudi pridobili sami s svojimi meritvami ali raziskavo. Obravnavani podatki so v `.csv` formatu.

Najprej jih naložimo z že znanimi ukazi:

```
pod <- read.table('./data_raw/possum.csv',  
                  header= T,  
                  sep = ",",  
                  dec = ".",  
                  quote = "\"\"")
```

Poglejmo, kako izgledajo podatki. Za to uporabimo funkcijo `head()`, ki izpiše začetnih `n` vrstic:

```
head(pod, 5)
```

```
##   case site Pop sex age hdlngth skullw totlngth taill footlgth earconch  eye  
## 1    1   1 Vic  m   8   94.1   60.4    89.0   36.0    74.5    54.5 15.2  
## 2    2   1 Vic  f   6   92.5   57.6    91.5   36.5    72.5    51.2 16.0  
## 3    3   1 Vic  f   6   94.0   60.0    95.5   39.0    75.4    51.9 15.5  
## 4    4   1 Vic  f   6   93.2   57.1    92.0   38.0    76.1    52.2 15.2  
## 5    5   1 Vic  f   2   91.5   56.3    85.5   36.0    71.0    53.2 15.1  
##   chest belly  
## 1   28.0    36  
## 2   28.5    33  
## 3   30.0    34  
## 4   28.0    34  
## 5   28.5    33
```

Ko vemo, kako so podatki strukturirani, lahko uporabimo funkcijo `summary()`, da pridobimo nekaj osnovnih značilnosti podatkov:

```
summary(pod)
```

```
##           case           site           Pop           sex  
##  Min.      : 1.00    Min.      :1.000    Length:104      Length:104  
## 1st Qu.: 26.75    1st Qu.:1.000    Class :character  Class :character
```

Ugotovimo, da imamo v podatkih nekaj manjkajočih vrednosti. Večina stolpcev je numeričnih, razen stolpcev **Pop** in **sex**, ki sta tipa **character**. Sumimo tudi, da spremenljivki **case** in **site** predstavljata kategorijo zato jih ne obravnavamo kot številske vrednosti. Poglejmo si, kako izgledajo vrednosti v stolpcih **case** in **site** (izpišemo prvih 50 vrednosti).

[illegible]

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

Poglejmo še podrobneje, katere vrednosti zavzemata. Pri tem si pomagamo s funkcijo `unique()`, ki nam izpiše edinstvene vrednosti v vektorju:

```
unique(pod$case)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
## [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
## [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
## [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
## [91] 91 92 93 94 95 96 97 98 99 100 101 102 103 104
```

```
unique(pod$site)
```

```
## [1] 1 2 3 4 5 6 7
```

Odločimo se, da spremenljivki za našo analizo nista pomembni in ju v nadaljevanju odstranimo. To storimo tako, da naše podatke “prepišemo” s podatki, ki ne vsebujejo stolpcev **case** in **site**.

```
pod <- pod[,setdiff(names(pod), c("case","site"))]
```

Poglejmo si še stolpca **Pop** in **sex**, ki vsebujeta nize:

```
pod$Pop
```

```
## [1] "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic"
## [10] "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic"
## [19] "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic"
## [28] "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic"
## [37] "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic" "Vic"
## [46] "Vic" "other" "other" "other" "other" "other" "other" "other" "other"
## [55] "other" "other" "other" "other" "other" "other" "other" "other" "other"
## [64] "other" "other" "other" "other" "other" "other" "other" "other" "other"
## [73] "other" "other" "other" "other" "other" "other" "other" "other" "other"
## [82] "other" "other" "other" "other" "other" "other" "other" "other" "other"
## [91] "other" "other" "other" "other" "other" "other" "other" "other" "other"
## [100] "other" "other" "other" "other" "other"
```

```
pod$sex
```

```
## [1] "m" "f" "f" "f" "f" "f" "m" "f" "f" "f" "f" "f" "m" "m" "m" "m" "f" "m"
## [19] "f" "f" "f" "m" "f" "m" "m" "m" "f" "m" "f" "f" "m" "f" "m" "m" "m" "m"
## [37] "f" "m" "f" "f" "f" "m" "f" "m" "m" "m" "m" "m" "m" "f" "f" "m" "f" "m"
## [55] "m" "m" "f" "m" "m" "f" "m" "f" "f" "f" "f" "f" "m" "m" "m" "f" "m" "m"
## [73] "m" "f" "m" "m" "m" "m" "m" "m" "m" "f" "f" "m" "m" "f" "m" "f" "m" "m"
## [91] "m" "m" "m" "m" "m" "m" "m" "m" "f" "m" "m" "f" "m" "f"
```

Stolpec **Pop** vsebuje podobno informacijo kot **site**, kjer vrednost *Vic* predstavlja območje Avstralije *Victoria*, oziroma vrednosti 1 in 2 v stolpcu **site** predstavljata verjetno dve področji v Victoriji. **other** pa predstavlja lokaciji *New South Wales* in *Queensland* v stolpcu **Pop**. Te vrednosti so v stolpcu **site** zakodirane z vrednostmi od 3 do 7. V naši raziskavi nas lokacija oposumov ne zanimajo, zato stolpec odstranimo.

```
pod <- pod[ , -which(colnames(pod)=="Pop")]
```

Še enkrat si pogledjmo osnovne statistike izbranih podatkov.

```
summary(pod)
```

```
##      sex          age      hdlngth      skullw
## Length:104      Min.   :1.000      Min.   : 82.50      Min.   :50.00
## Class :character 1st Qu.:2.250      1st Qu.: 90.67      1st Qu.:54.98
## Mode  :character Median :3.000      Median : 92.80      Median :56.35
##                               Mean  :3.833      Mean   : 92.60      Mean   :56.88
##                               3rd Qu.:5.000      3rd Qu.: 94.72      3rd Qu.:58.10
##                               Max.   :9.000      Max.   :103.10      Max.   :68.60
##                               NA's   :2
##      totlngth      taill      footlngth      earconch
## Min.   :75.00      Min.   :32.00      Min.   :60.30      Min.   :40.30
## 1st Qu.:84.00      1st Qu.:35.88      1st Qu.:64.60      1st Qu.:44.80
## Median :88.00      Median :37.00      Median :68.00      Median :46.80
## Mean   :87.08      Mean   :37.01      Mean   :68.46      Mean   :48.13
## 3rd Qu.:90.00      3rd Qu.:38.00      3rd Qu.:72.50      3rd Qu.:52.00
## Max.   :96.50      Max.   :43.00      Max.   :77.90      Max.   :56.20
## NA's   :7          NA's   :1
##      eye      chest      belly
## Min.   :12.80      Min.   :22.0      Min.   :25.00
## 1st Qu.:14.40      1st Qu.:25.5      1st Qu.:31.00
## Median :14.90      Median :27.0      Median :32.50
## Mean   :15.05      Mean   :27.0      Mean   :32.59
## 3rd Qu.:15.72      3rd Qu.:28.0      3rd Qu.:34.12
## Max.   :17.80      Max.   :32.0      Max.   :40.00
##
```

Stolpec **sex** lahko zavzema dve vrednosti **m** in **f**, to sta edini možni vrednosti v teh podatkih. S pomočjo tipa **faktor** lahko zagotovimo, da ta stolpec vsebuje le vrednosti **m** in **f**.

```
pod$sex <- factor(pod$sex, levels = c("m", "f"))
```

Cilj današnje naloge je modelirati celotno dolžino oposuma (**totlngth**) v odvisnosti od velikosti njegove glave (**hdlngth**). Nekaj vrednosti celotne dolžine manjka, zato vrstice, kjer manjka vrednost za **totlngth** odstranimo. To so vrednosti, ki jih modeliramo, zato teh manjkajočih vrednosti ni primerno nadomeščati z povprečjem ali podobnimi tehnikami. Stolpec **hdlngth** ne vsebuje manjkajočih vrednosti.

Odstranjevanje vrstic izvedemo tako, da izberemo le vrstice, kjer vrednost za **totlngth** ni manjkajoča.

```
pod <- pod[!(is.na(pod$totlngth)), ]
```

Poiščimo še druge stolpce z manjkajočimi vrednostmi. To lahko naredimo tako, da našo tabelo pretvorimo v logično tabelo kjer vrednost **TRUE** označuje manjkajoče vrednosti.

```
pod_na <- is.na(pod)
```

V logični tabeli poiščemo število manjkajočih vrednosti za vsak stolpec.

```
num_na <- apply(pod_na, 2, sum)
```

Poiščemo indekse stolpcev, ki vsebujejo manjkajoče vrednosti.

```
na_ind <- which(num_na > 0)
print(na_ind)
```

```
##      age footlgth
##      2         7
```

Te manjkajoče podatke bomo “imputirali”, to pomeni, da jih bomo nadomestili z neko drugo vrednostjo. Numerične podatke ponavadi nadomeščamo s povprečno vrednostjo podatkov v stolpcu, medtem ko kategorične vrednosti nadomeščamo z najpogostejšo vrednostjo v izbranem stolpcu.

Ker so manjkajoče vrednosti v več stolpcih, jih bomo nadomestili kar z zanko. Vsak obhod zanke bo nadomestil vrednosti v enem stolpcu. (Premislite, če bi lahko to storili s funkcijo `apply`.)

Najprej bomo napisali funkcijo, ki bo v podanem stolpcu nadomestila manjkajoče vrednosti s povprečno vrednostjo stolpca. Funkcija bo sprejela `data.frame` in indeks stolpca. Manjkajoče vrednosti v izbranem stolpcu bo nadomestila s povprečno vrednostjo tistega stolpca. Funkcija izgledala tako:

```
nadomesti_manjkajoce <- function(pod, i){
  pod[is.na(pod[, i]), i] <- mean(pod[, i], na.rm = T)
  return(pod)
}
```

Manjkajoče vrednosti v različnih stolpcih nadomestimo v telesu `for` zanke. Vsak obhod zanke predstavlja posodobitev manjkajočih vrednosti enega stolpca.

```
for (i in na_ind){
  pod <- nadomesti_manjkajoce(pod, i)
}
```

Na koncu še enkrat preverimo podatke po vseh opravljenih korakih predobdelave:

```
summary(pod)
```

```
## sex      age      hdlngth      skullw      totlngth
## m:56  Min.   :1.000  Min.   : 84.70  Min.   :50.00  Min.   :75.00
## f:41  1st Qu.:3.000  1st Qu.: 90.70  1st Qu.:55.00  1st Qu.:84.00
##      Median :3.000  Median : 92.80  Median :56.40  Median :88.00
##      Mean   :3.853  Mean   : 92.66  Mean   :56.93  Mean   :87.08
##      3rd Qu.:5.000  3rd Qu.: 94.50  3rd Qu.:58.10  3rd Qu.:90.00
##      Max.   :9.000  Max.   :103.10  Max.   :68.60  Max.   :96.50
##      taill      footlgth      earconch      eye
## Min.   :32.00  Min.   :60.30  Min.   :40.30  Min.   :12.80
## 1st Qu.:35.50  1st Qu.:64.50  1st Qu.:44.80  1st Qu.:14.40
## Median :36.50  Median :68.00  Median :46.80  Median :14.90
## Mean   :36.97  Mean   :68.35  Mean   :48.02  Mean   :15.02
## 3rd Qu.:38.00  3rd Qu.:72.30  3rd Qu.:52.00  3rd Qu.:15.70
## Max.   :43.00  Max.   :77.90  Max.   :56.20  Max.   :17.80
##      chest      belly
```

```
## Min.      :22.00   Min.      :25.00
## 1st Qu.:25.50   1st Qu.:31.00
## Median :27.00   Median :32.00
## Mean    :27.03   Mean    :32.54
## 3rd Qu.:28.00   3rd Qu.:34.00
## Max.     :32.00   Max.     :40.00
```

Risanje odvisnosti med ciljno spremenljivko in ostalimi spremenljivkami

Sedaj narišimo odvisnosti med ciljno spremenljivko **totlngh** in ostalimi spremenljivkami. To naredimo s pomočjo knjižnice **ggplot2**, tako kot smo se naučili na petem predavanju.

Najprej naložimo knjižnici za risanje podatkov (**ggplot2**) in knjižnico (**tidyr**), ki vsebuje tudi funkcije za pretvarjanje podatkov iz široke v dolgo obliko.

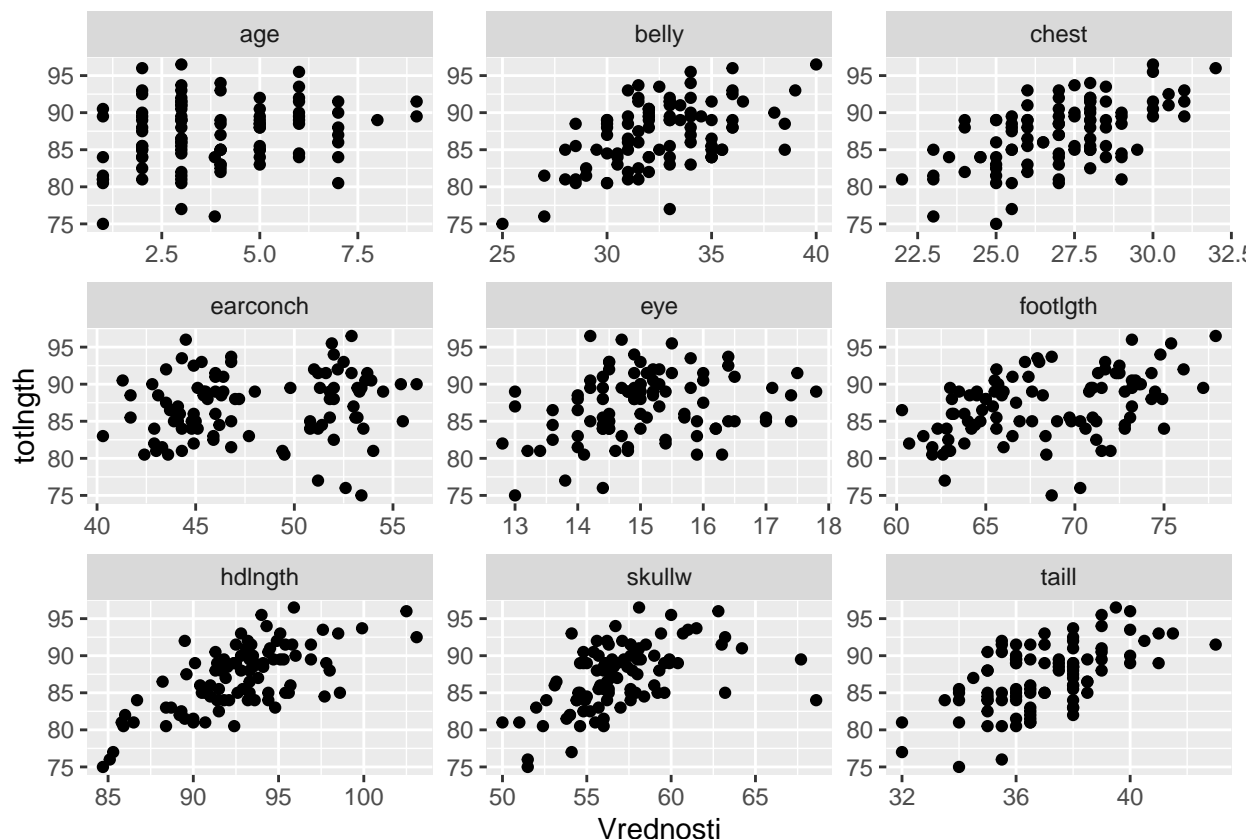
```
library(ggplot2)
library(tidyr)
```

Izberemo le numerične podatke in jih pripravimo za vizualizacijo s paketom **ggplot2**. Obliko tabele spremenimo iz široke v dolgo s pomočjo paketa **tidy**.

```
pod_n <- pod[ , -which(names(pod) == "sex")]
pod_n <- pivot_longer(pod_n, cols = setdiff(names(pod_n), "totlngh"),
                      names_to = "Spremenljivke", values_to = "Vrednosti")
```

Narišemo odvisnosti s pomočjo razsevnega diagrama. S funkcijo **facet_wrap()** lahko narišemo vse odvisnosti hkrati. Funkcija **facet_wrap()** ustvari toliko grafov, kolikor je odvisnosti.

```
ggplot(pod_n, aes(Vrednosti, totlngh)) +
  geom_point() +
  facet_wrap(~Spremenljivke, scales = "free")
```



Opazimo, da **totlngth** izkazuje dokaj linearno relacijo z drugimi spremenljivkami, ki opisujejo velikost (npr. dolžina repa: **taill**, obseg trebuha: **belly**, obseg prsi: **chest** in tudi dolžina glave: **hdlngth**).

Linearni model

Naš končni cilj je preveriti, če se da oceniti dolžino telesa oposuma iz dolžine njegove glave. Za to bomo uporabili linearni model. Rezultat linearne regresije je enačba premice:

```
model <- lm(totlngth ~ hdlngth, data = pod)
```

Več podatkov o modelu dobimo s funkcijo **summary**:

```
summary(model)
```

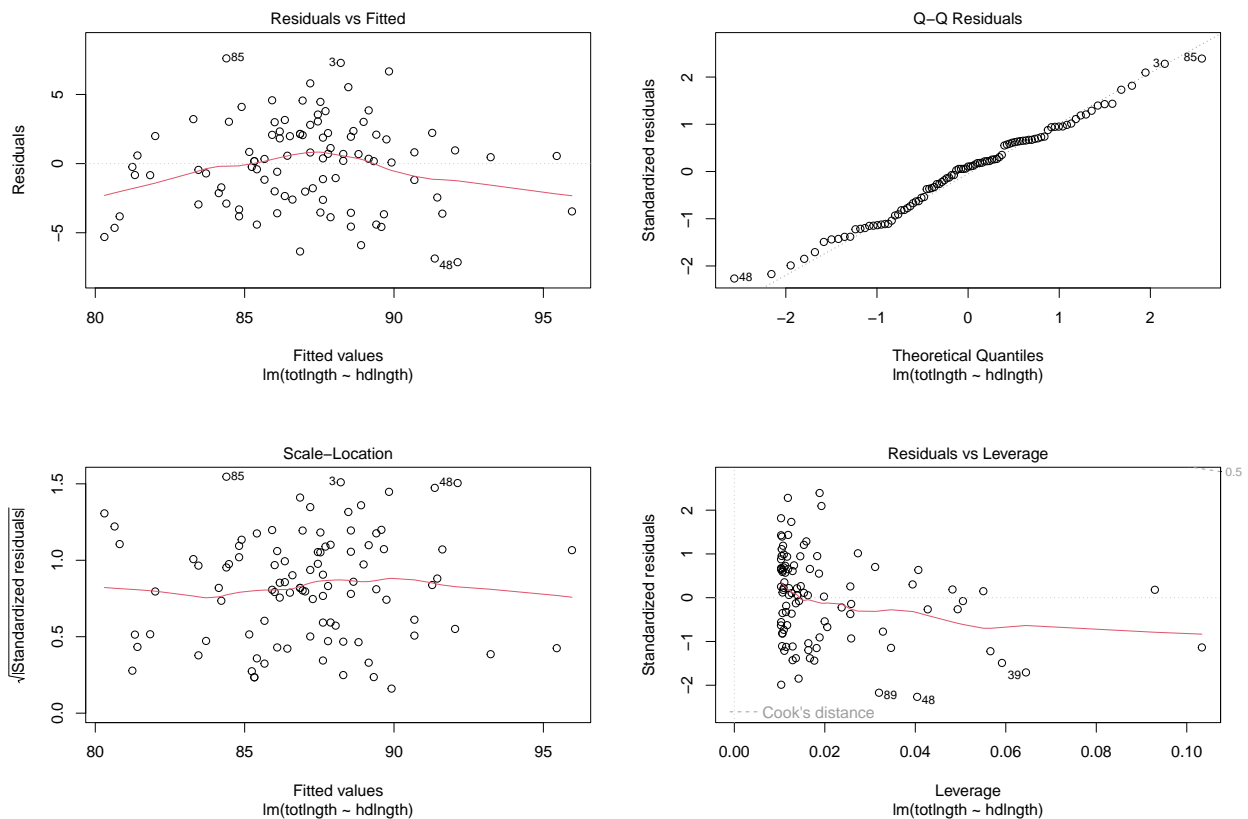
```
##
## Call:
## lm(formula = totlngth ~ hdlngth, data = pod)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.1297 -2.4492  0.3351  2.1442  7.6111
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 8.25758 8.70551 0.949 0.345
## hdlngth     0.85063 0.09388 9.061 1.68e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.213 on 95 degrees of freedom
## Multiple R-squared:  0.4636, Adjusted R-squared:  0.4579
## F-statistic: 82.09 on 1 and 95 DF,  p-value: 1.683e-14
```

Funkcija `summary()` nam vrne veliko informacij o dobljenem linearnem modelu. Dolžina glave je označena z *******, kar nakazuje, da je dolžina glave signifikantno linearno povezana z dolžino celotnega telesa. Vrednost R^2 nam prikazuje, kolikšen del variance v podatkih o dolžini teles razloži naš linearen model.

Različne grafe, ki nam pomagajo diagnosticirati linearno regresijo, preprosto dobimo z funkcijo `plot()`:

```
plot(model)
```



Sedaj želimo pogledati, še kako vse ostale spremenljivke skupaj vplivajo na dolžino telesa.

```
model2 <- lm(totlngth ~ ., data = pod)
summary(model2)
```

```
##
## Call:
## lm(formula = totlngth ~ ., data = pod)
```



```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.6069 -1.3472  0.2082  1.3499  5.6990
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -30.235584   8.413151  -3.594 0.000542 ***
## sexf         1.294747   0.509106   2.543 0.012774 *
## age        -0.082845   0.135356  -0.612 0.542116
## hdlngth     0.504903   0.111654   4.522 1.95e-05 ***
## skullw     -0.007117   0.113281  -0.063 0.950050
## taill       1.143191   0.142866   8.002 5.24e-12 ***
## footlngth   0.158183   0.099327   1.593 0.114929
## earconch    0.178710   0.104360   1.712 0.090421 .
## eye         0.143650   0.252600   0.569 0.571053
## chest       0.187661   0.178487   1.051 0.296021
## belly       0.055761   0.114174   0.488 0.626519
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.284 on 86 degrees of freedom
## Multiple R-squared:  0.7546, Adjusted R-squared:  0.726
## F-statistic: 26.44 on 10 and 86 DF, p-value: < 2.2e-16
```

Ugotovimo, da nam dolžina glave in dolžina repa največ povesta o dolžini celotnega oposuma. Nekaj informacije o dolžini nam pove tudi spol oposuma in kako velika ušesa ima (**earconch**). Tak model razloži veliko več variance v podatkih o dolžini oposuma, saj je R^2 tukaj 0.73.

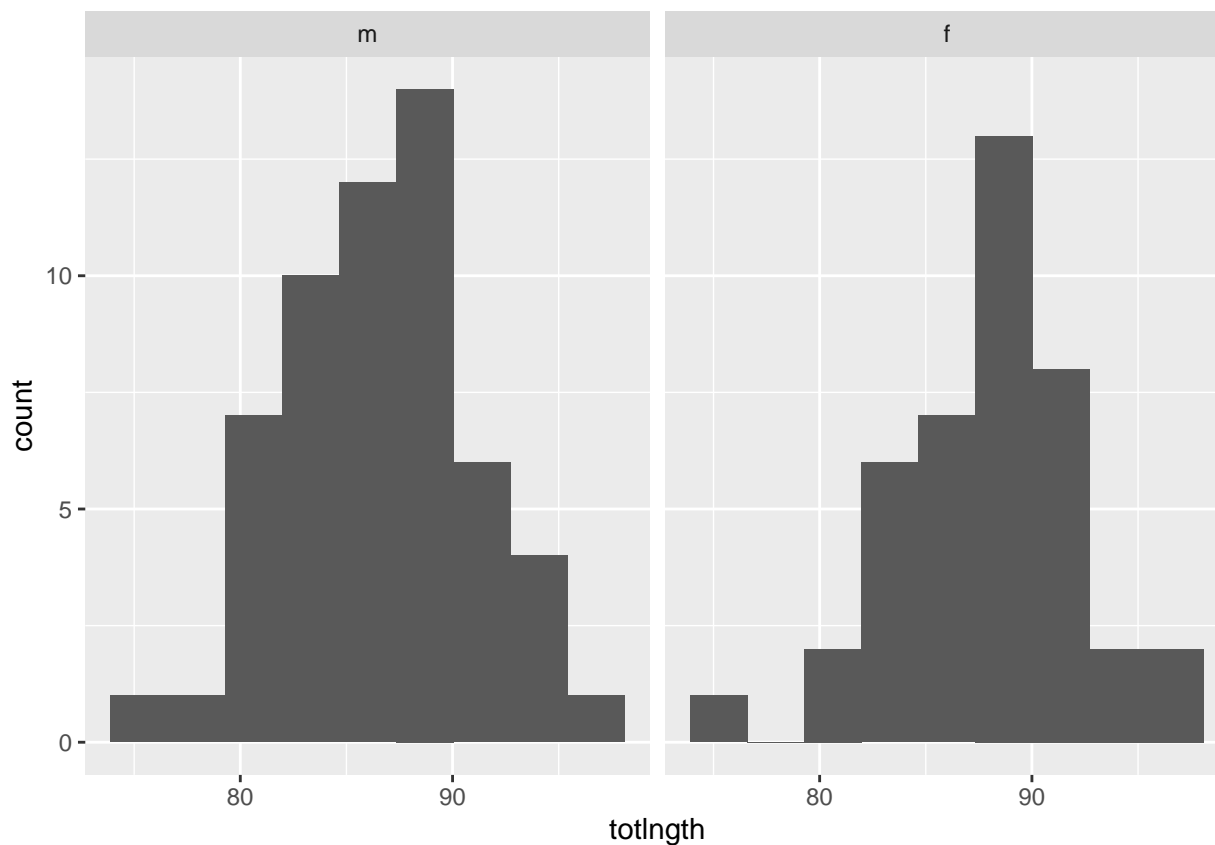
Statistična signifikantnost – p vrednosti

Večkrat želimo preveriti, če je neka razlika v podatkih statistično signifikantna. Za to ponavadi uporabimo t-test, ki predpostavlja normalno porazdelitev. Zato najprej pogledamo, kako so podatki celotne velikosti oposumov porazdeljeni po spolih.

Narišimo histograme dolžin za moške in ženske oposume.

Izberemo podatke o spolu in celotni dolžini in jih pretvorimo v dolgo obliko

```
pod_dolzina <- pod[, colnames(pod) == 'sexf' | colnames(pod) == 'totlngth']
ggplot(pod_dolzina, aes(x=totlngth)) +
  geom_histogram(bins = 9) +
  facet_wrap(~sexf)
```



Opazimo, da so podatki normalno porazdeljeni glede na spol. Podatkov imamo sicer malo, ampak na splošno lahko sklepamo, da so podatki o velikosti za posamezen spol normalno porazdeljeni.

Sedaj uporabimo t-test, da preverimo, če so razlike v velikosti oposumov med spoloma statistično signifikantne.

```
dolzine_f <- pod_dolzina[pod_dolzina$sex == "f", ]$totlngth
dolzine_m <- pod_dolzina[pod_dolzina$sex == "m", ]$totlngth
t.test(dolzine_f,
       dolzine_m)
```

```
##
## Welch Two Sample t-test
##
## data:  dolzine_f and dolzine_m
## t = 1.4161, df = 88.016, p-value = 0.1603
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5069178  3.0202453
## sample estimates:
## mean of x mean of y
##  87.80488  86.54821
```

Ugotovimo, da razlike niso statistično signifikantne.