Correlation and Regression

Valeria Bogdanova

Data Analyst at Semrush

Spring 2021

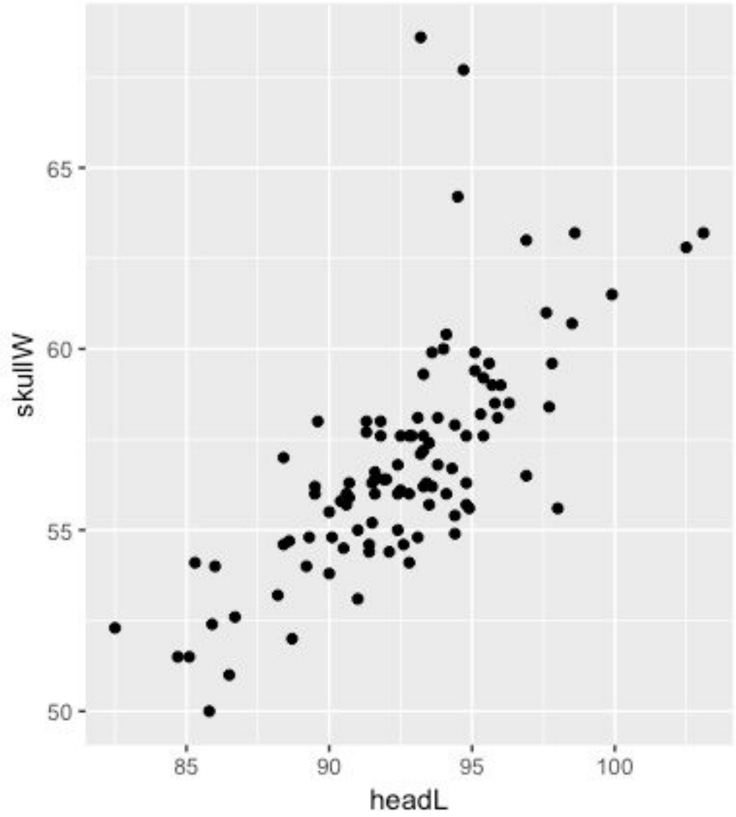
Bivariate Relationships

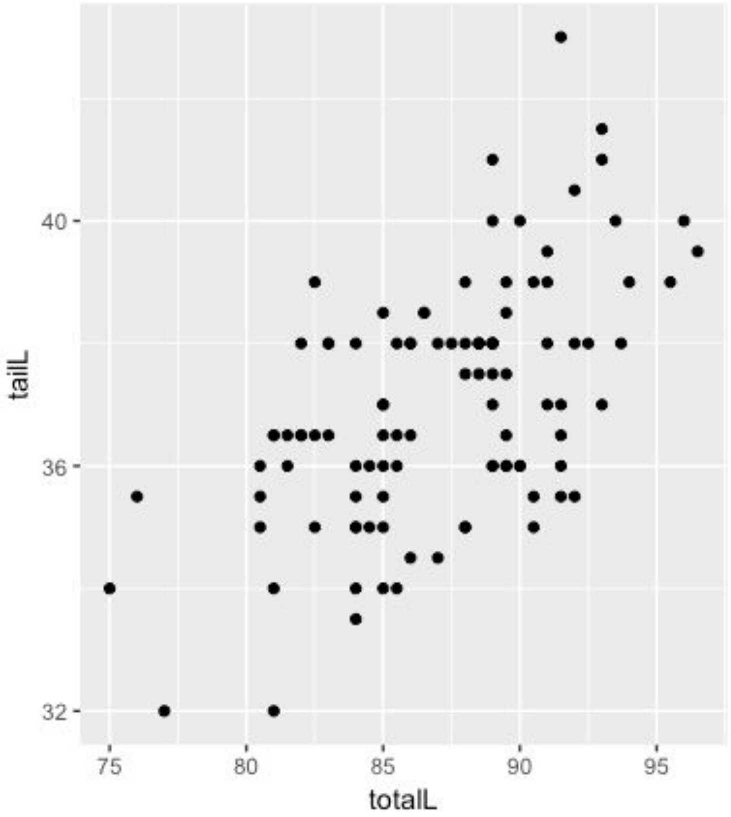
- Two numerical variables: y ~ x

- y — dependent, response
- x — independent, explanatory, predictor

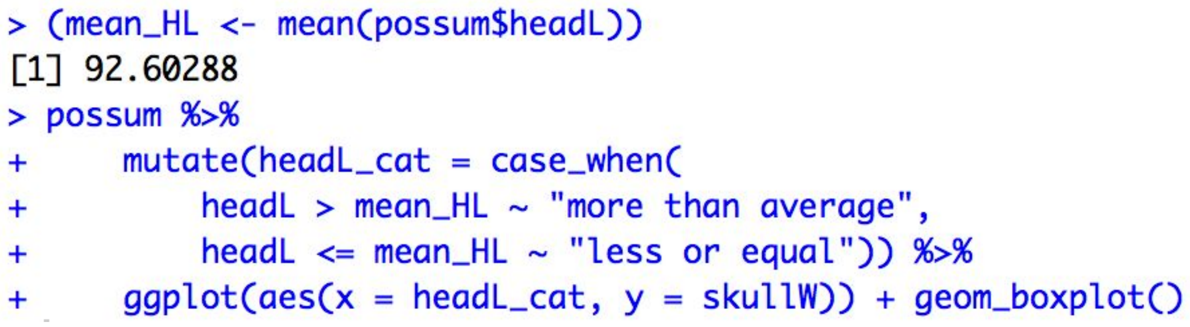
Graphical Exploration

- Scatter plot
- In case of need — box plot

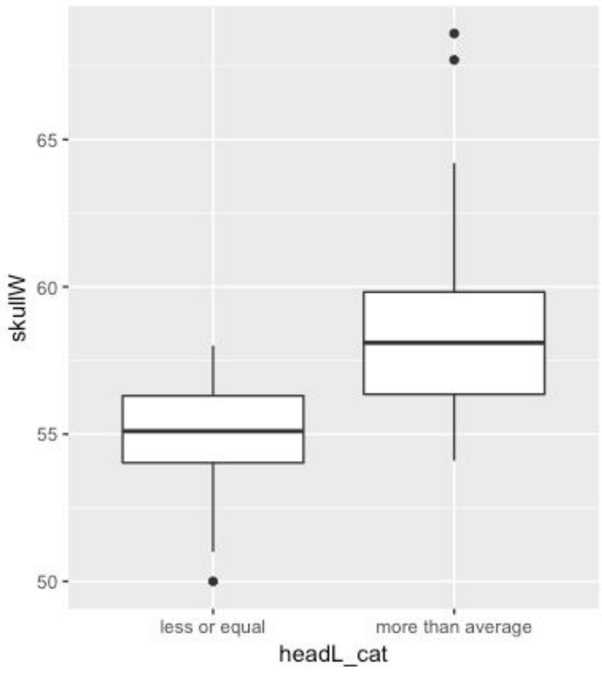


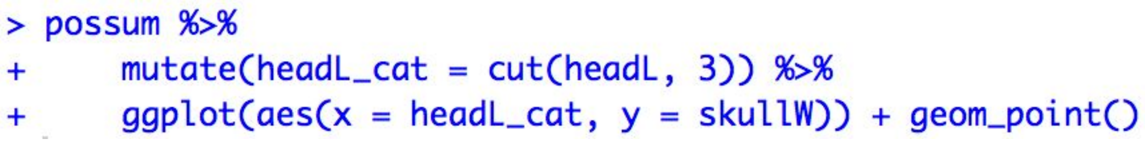


Graphical Exploration

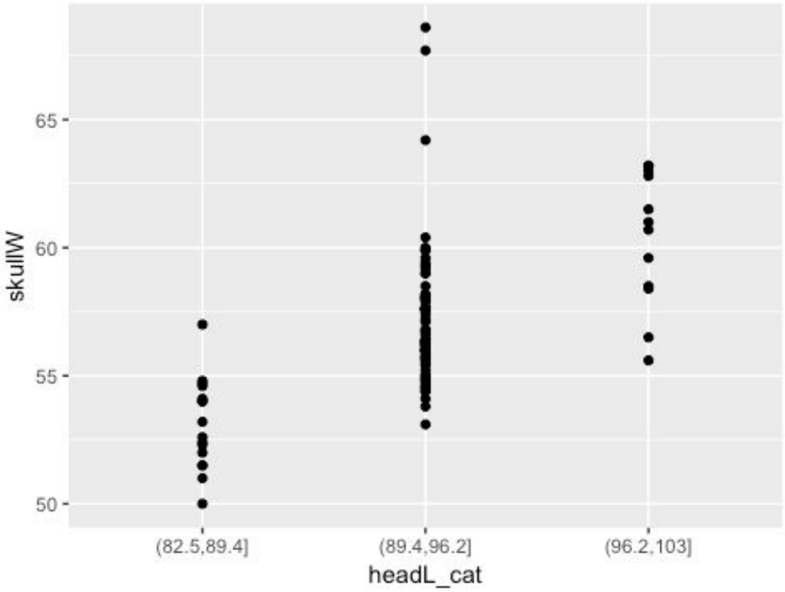


```
> (mean_HL <- mean(possum$headL))
[1] 92.60288
> possum %>%
+     mutate(headL_cat = case_when(
+         headL > mean_HL ~ "more than average",
+         headL <= mean_HL ~ "less or equal")) %>%
+     ggplot(aes(x = headL_cat, y = skullW)) + geom_boxplot()
```





```
> possum %>%
+     mutate(headL_cat = cut(headL, 3)) %>%
+     ggplot(aes(x = headL_cat, y = skullW)) + geom_point()
```

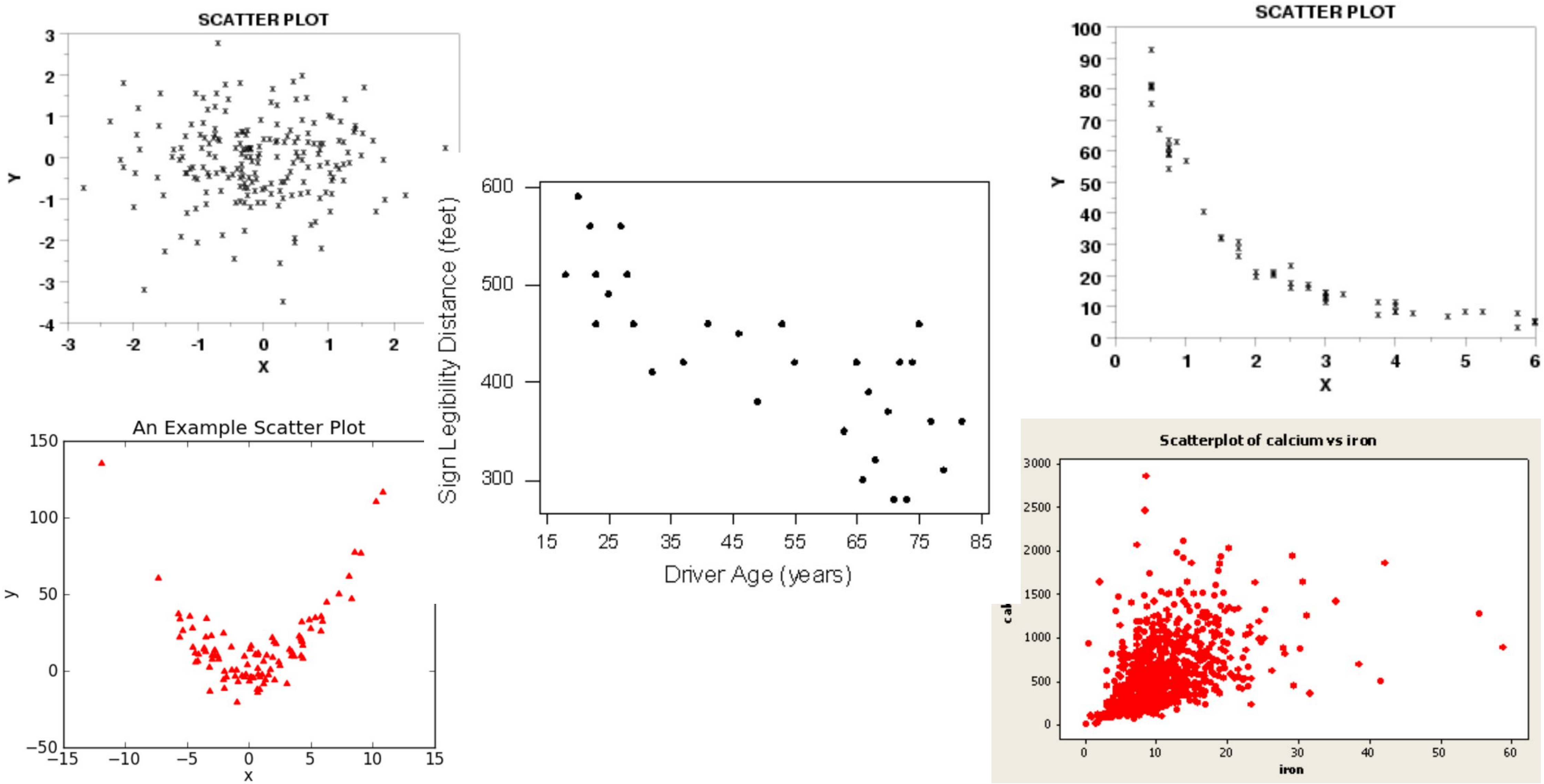


Relationship Characteristics

- **Form**: linear / non-linear
- **Direction**: positive, negative
- **Strength**: weak, moderate, strong

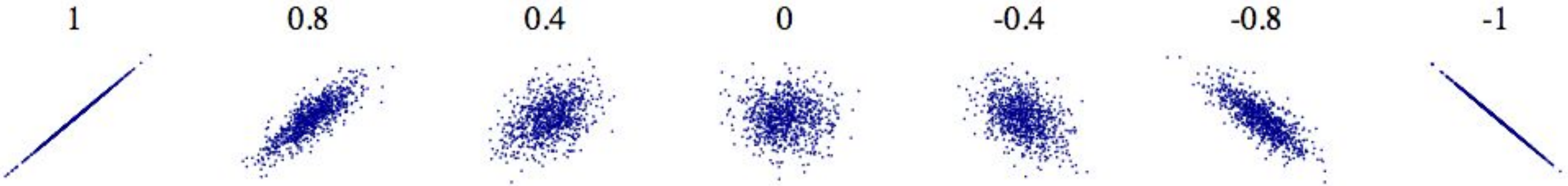
- Outliers

Relationship Characteristics

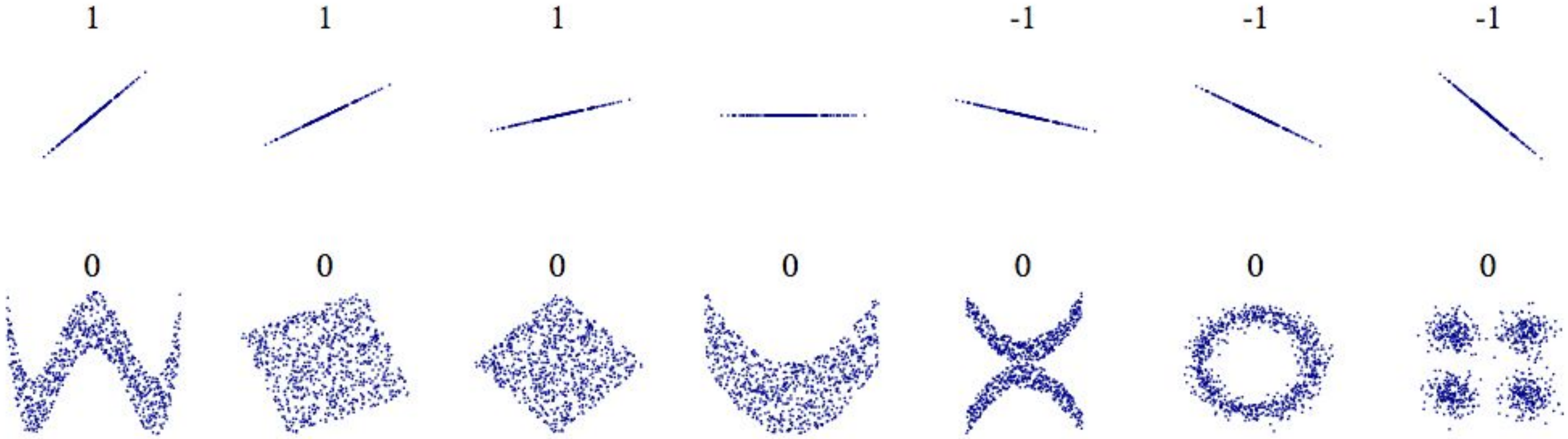


Correlation (Pearson’s)

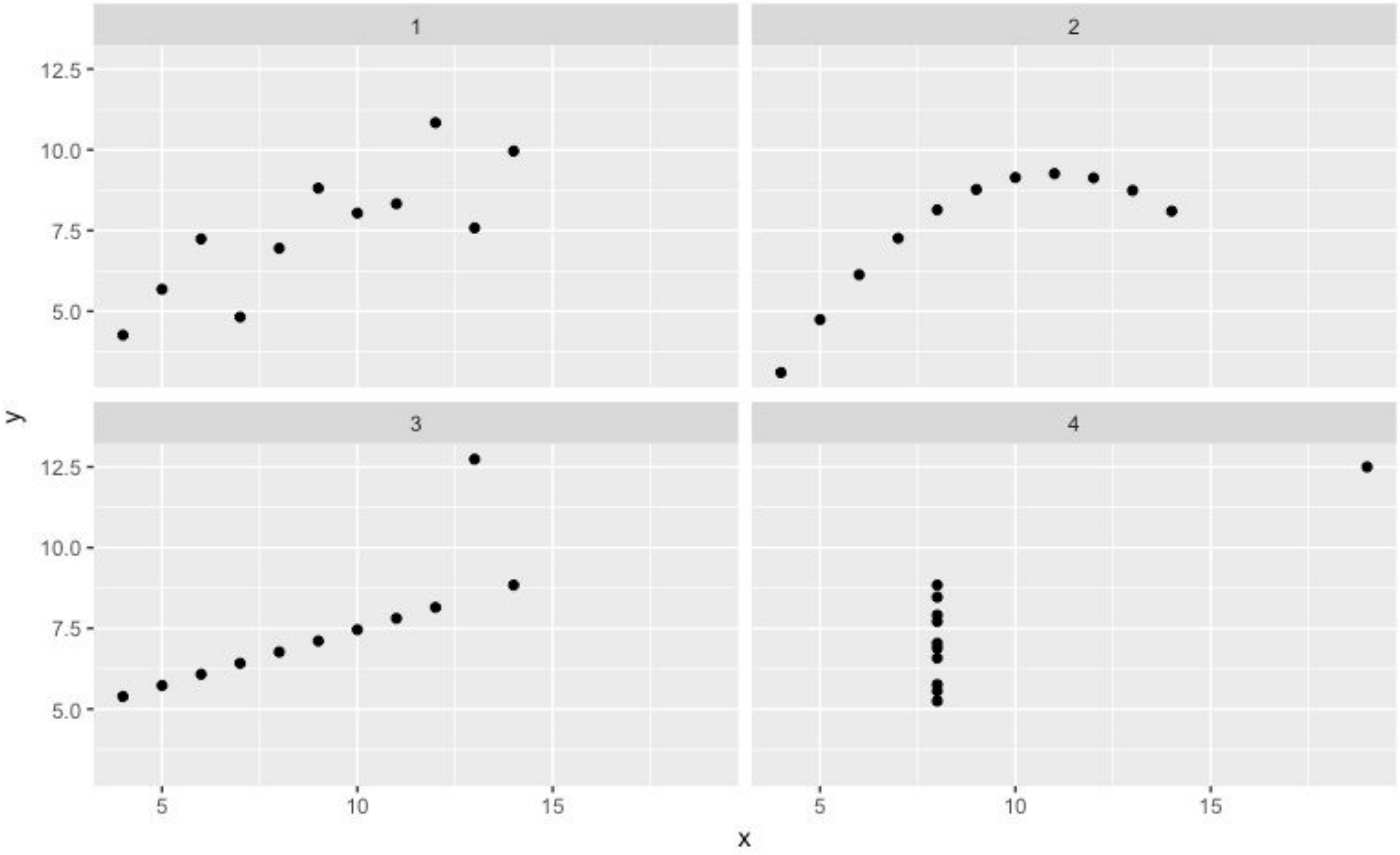
$$\mathbf{r}_{XY} = \frac{\mathbf{cov}_{XY}}{\sigma_X \sigma_Y} = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2 \sum(Y - \bar{Y})^2}}$$

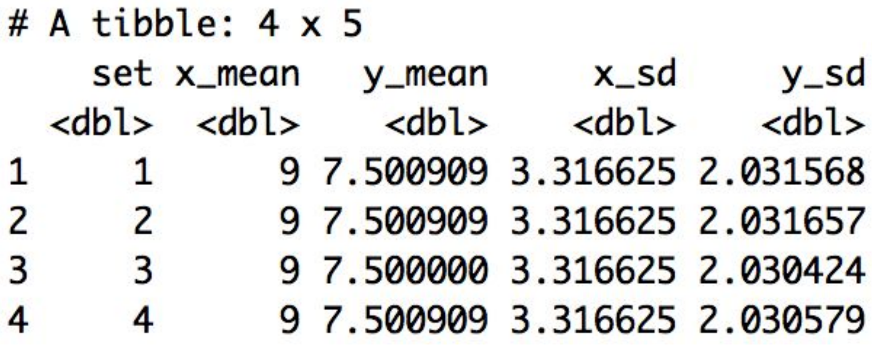


Correlation (Pearson's)

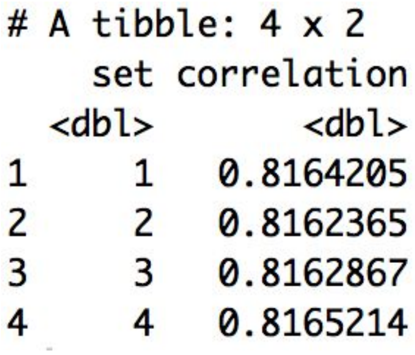


Task: Anscombe’s Quartet



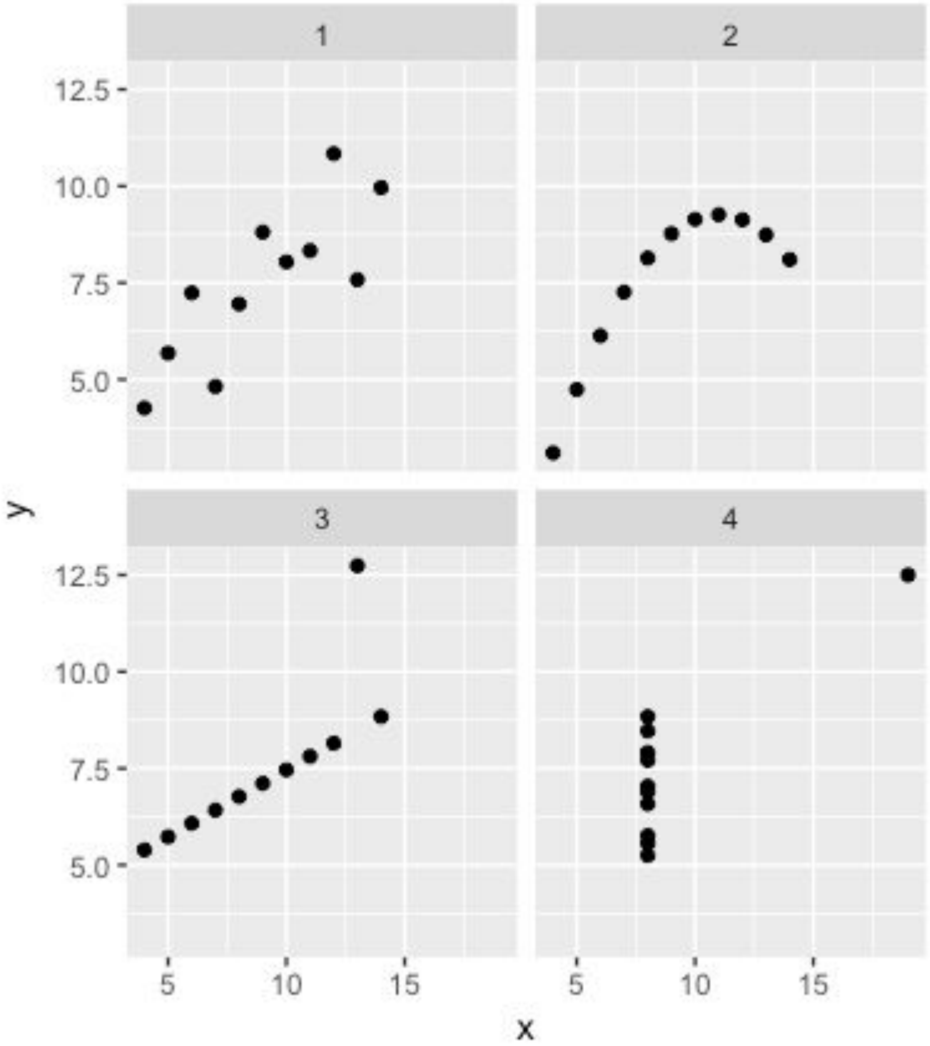


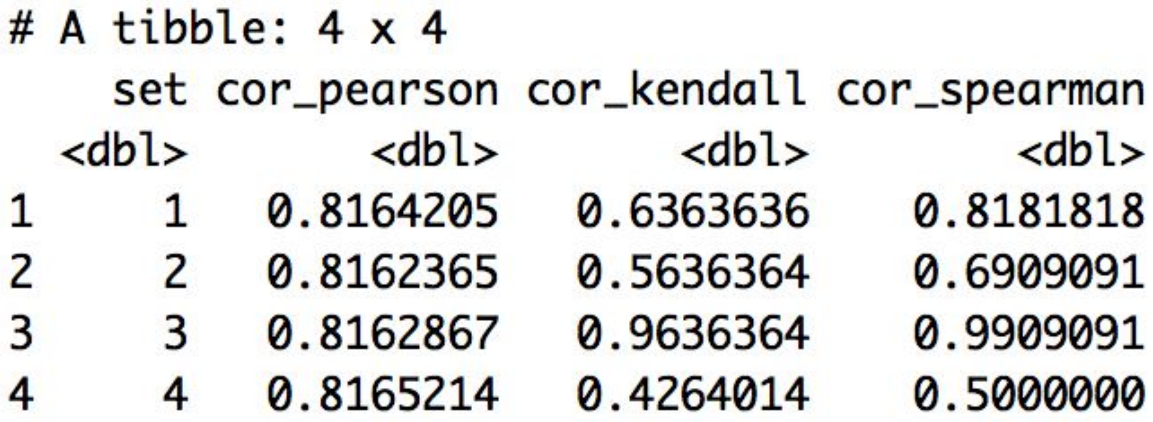
```
# A tibble: 4 x 5
    set x_mean   y_mean     x_sd     y_sd
  <dbl>  <dbl>    <dbl>    <dbl>    <dbl>
1     1      9 7.500909 3.316625 2.031568
2     2      9 7.500909 3.316625 2.031657
3     3      9 7.500000 3.316625 2.030424
4     4      9 7.500909 3.316625 2.030579
```



```
# A tibble: 4 x 2
    set correlation
  <dbl>       <dbl>
1     1   0.8164205
2     2   0.8162365
3     3   0.8162867
4     4   0.8165214
```

Task: Correlation





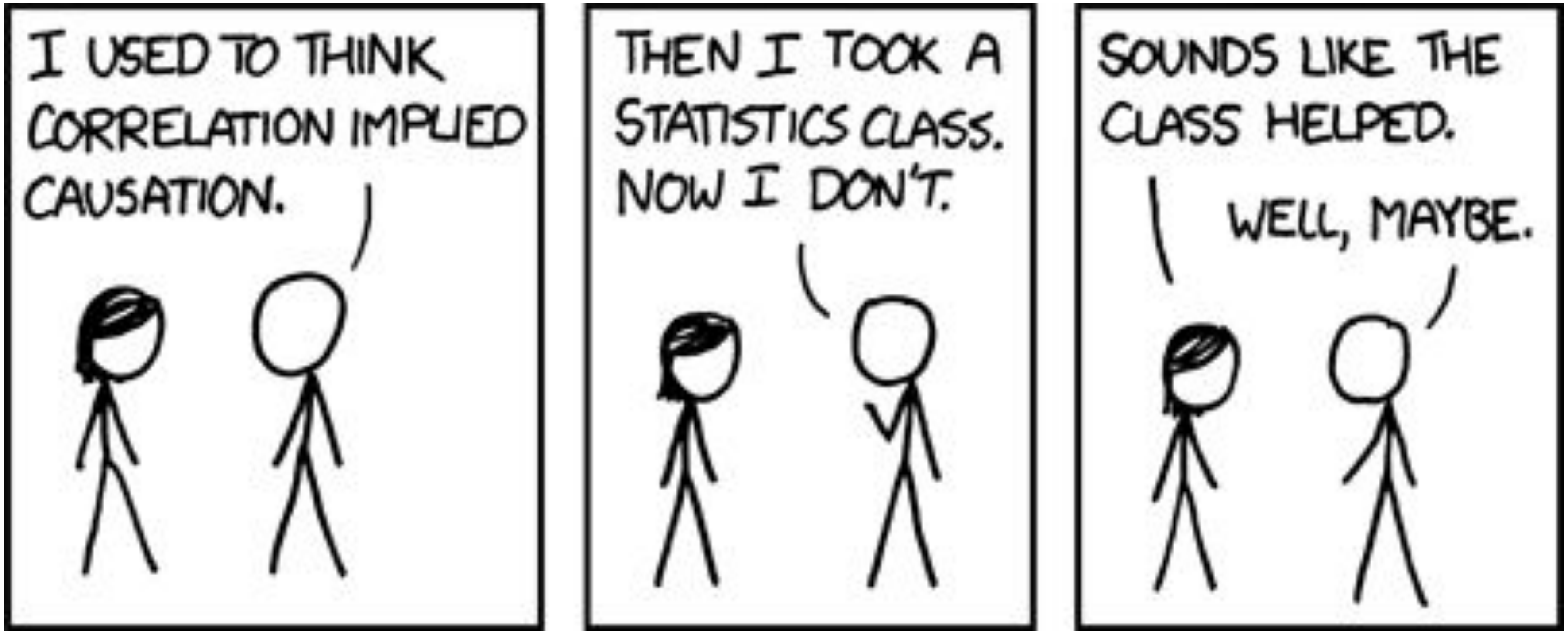
```
# A tibble: 4 x 4
    set cor_pearson cor_kendall cor_spearman
  <dbl>       <dbl>       <dbl>        <dbl>
1     1   0.8164205   0.6363636    0.8181818
2     2   0.8162365   0.5636364    0.6909091
3     3   0.8162867   0.9636364    0.9909091
4     4   0.8165214   0.4264014    0.5000000
```

Correlation

- Significance

```
cor.test(x, y)$p.value
```

- Task: add p-values

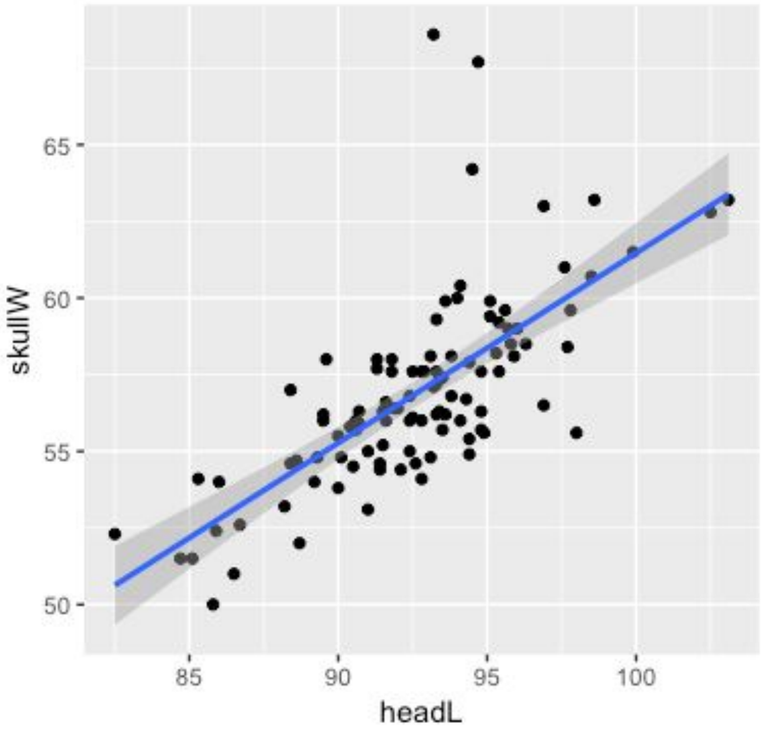

I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

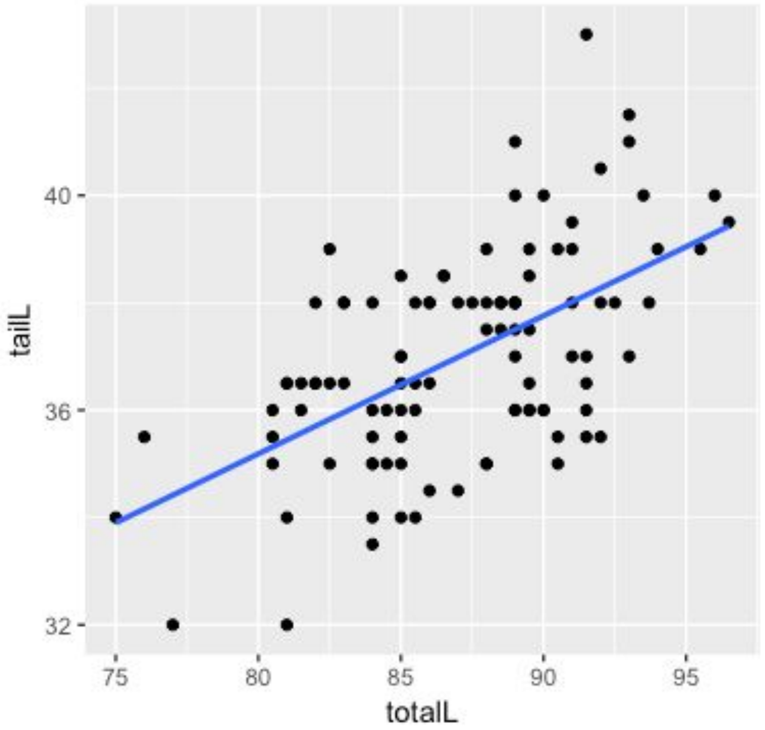
https://xkcd.com/552/

Linear Regression

```r
possum %>%
    ggplot(aes(x = headL, y = skullW)) +
    geom_point() +
    geom_smooth(method = "lm")
```

```r
possum %>%
    ggplot(aes(x = totalL, y = tailL)) +
    geom_point() +
    geom_smooth(method = "lm", se = F)
```





Linear Regression

response = f(explanatory) + noise

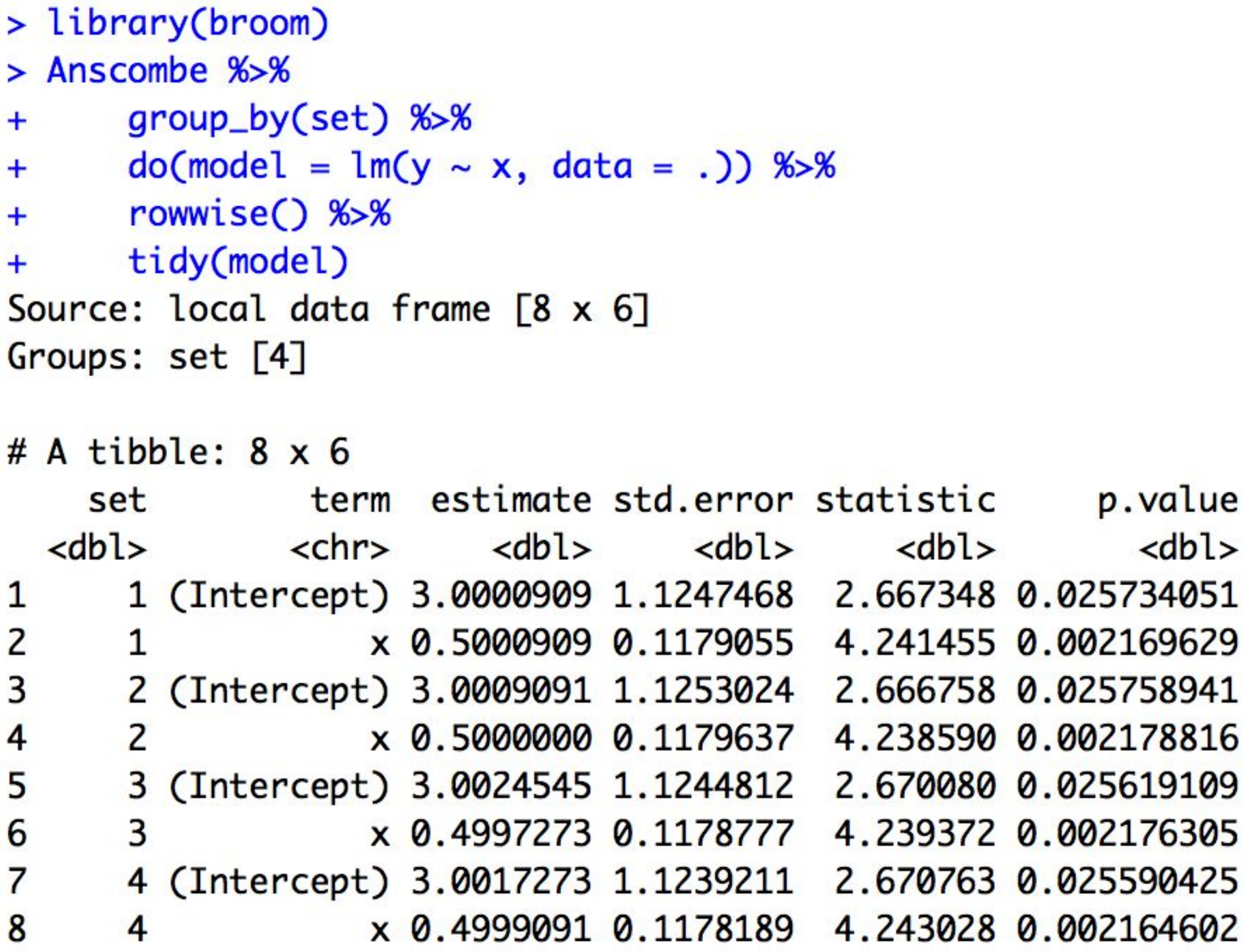
*response = intercept + (slope * explanatory) + noise*

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon\,, \qquad \epsilon \sim N(0, \sigma_\epsilon)$$

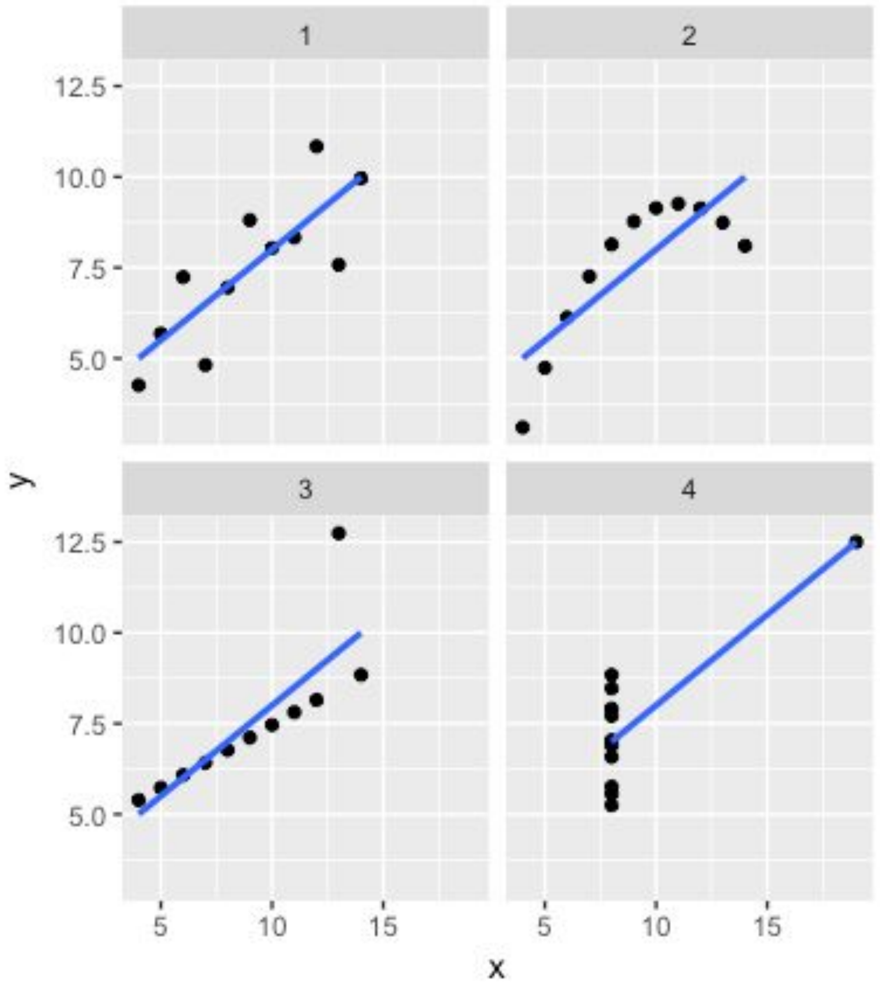
$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot X \qquad e = Y - \hat{Y}$$

- Given n observations of pairs (x_i, y_i)…
- Find $\hat{\beta}_0, \hat{\beta}_1$ that minimize $\sum_{i=1}^{n} e_i^2$

Linear Regression



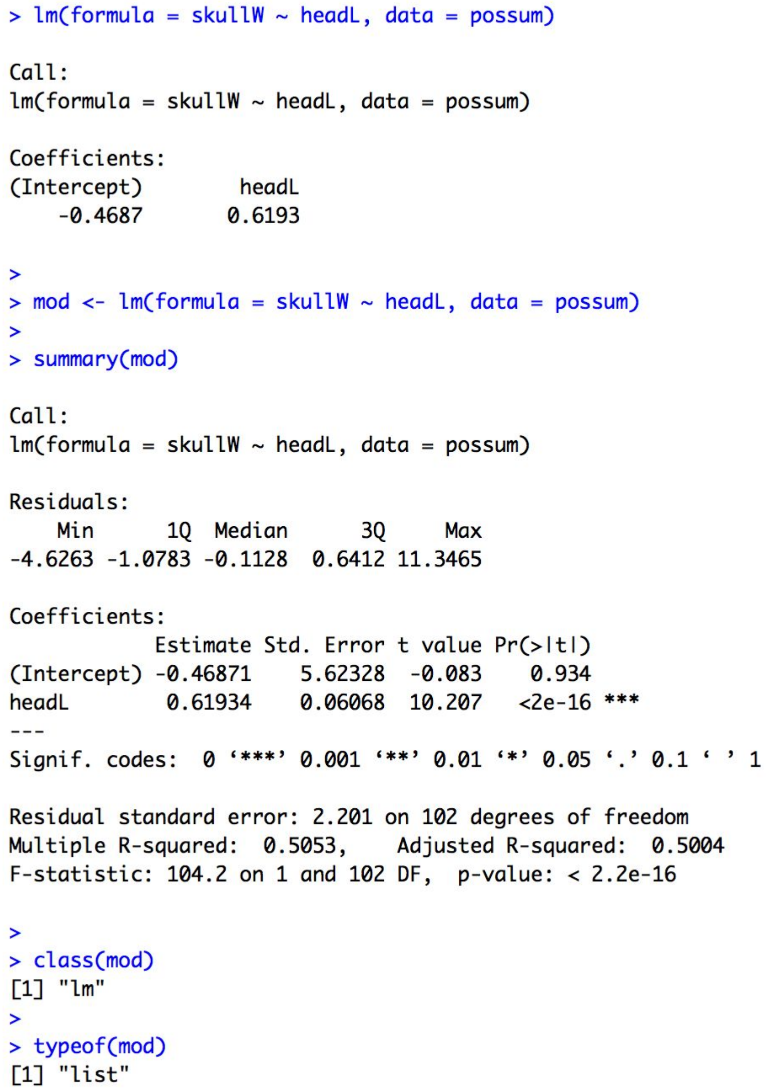
```
> library(broom)
> Anscombe %>%
+     group_by(set) %>%
+     do(model = lm(y ~ x, data = .)) %>%
+     rowwise() %>%
+     tidy(model)
Source: local data frame [8 x 6]
Groups: set [4]

# A tibble: 8 x 6
    set        term  estimate std.error statistic     p.value
  <dbl>       <chr>     <dbl>     <dbl>     <dbl>       <dbl>
1     1 (Intercept) 3.0000909 1.1247468  2.667348 0.025734051
2     1           x 0.5000909 0.1179055  4.241455 0.002169629
3     2 (Intercept) 3.0009091 1.1253024  2.666758 0.025758941
4     2           x 0.5000000 0.1179637  4.238590 0.002178816
5     3 (Intercept) 3.0024545 1.1244812  2.670080 0.025619109
6     3           x 0.4997273 0.1178777  4.239372 0.002176305
7     4 (Intercept) 3.0017273 1.1239211  2.670763 0.025590425
8     4           x 0.4999091 0.1178189  4.243028 0.002164602
```



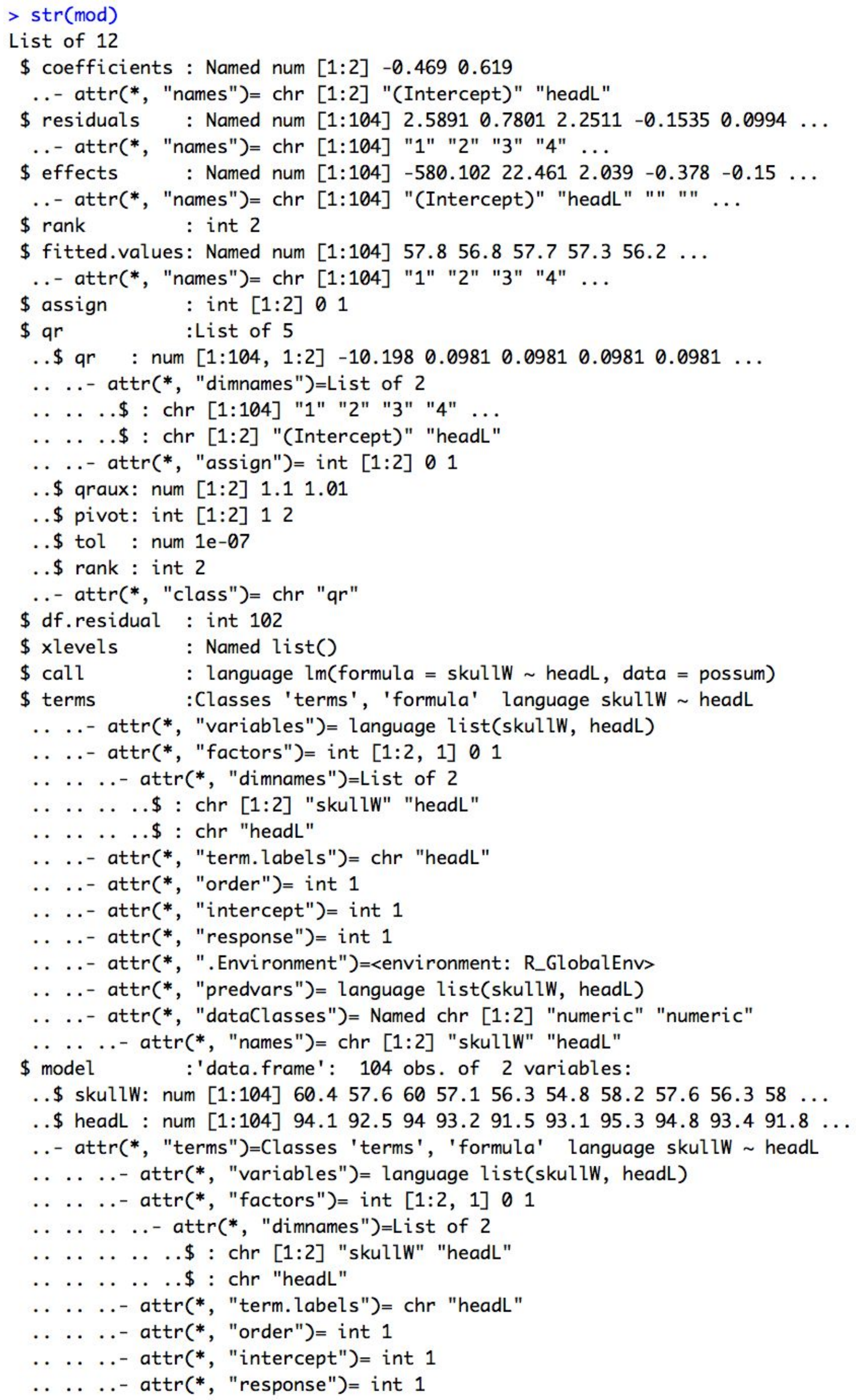
Linear Regression

- Assumptions:
  - Linear dependency between response and predictor
  - Constant variance (a.k.a. homoscedasticity)
  - Errors are normally distributed and independent

Linear Regression

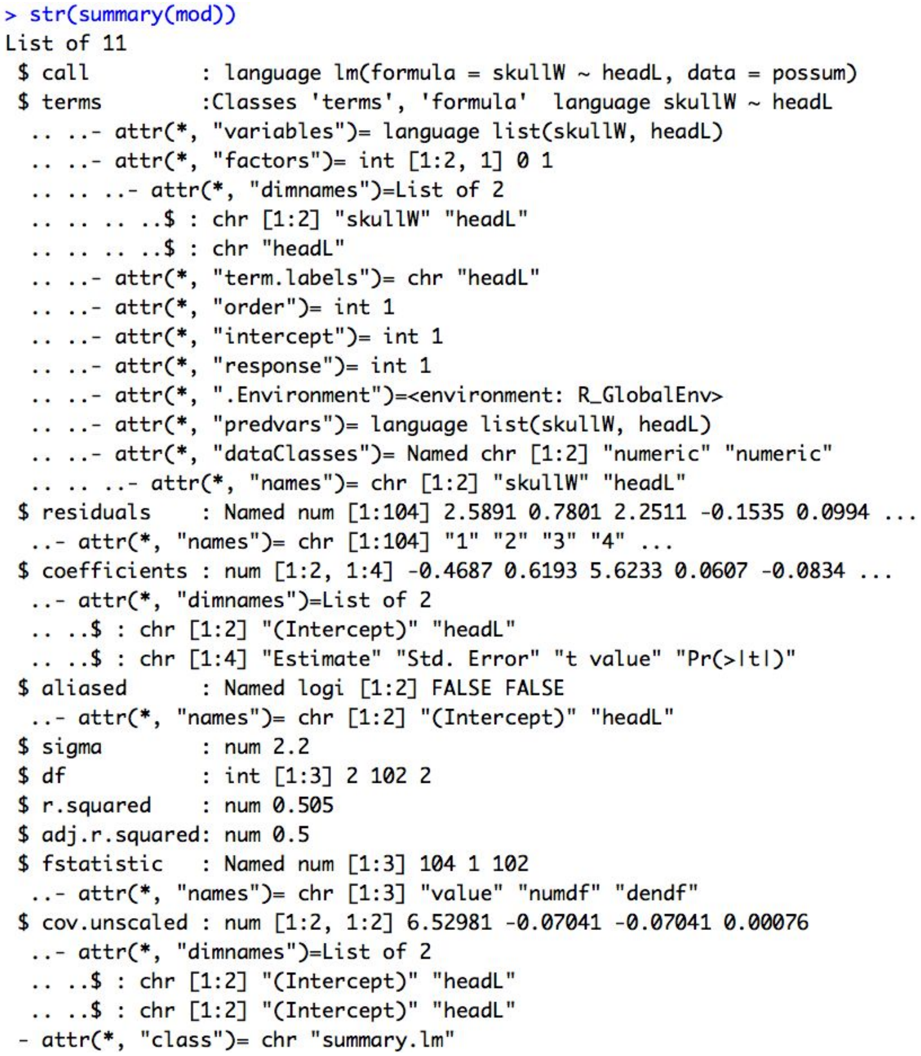


```
> lm(formula = skullW ~ headL, data = possum)

Call:
lm(formula = skullW ~ headL, data = possum)

Coefficients:
(Intercept)        headL
    -0.4687       0.6193

>
> mod <- lm(formula = skullW ~ headL, data = possum)
>
> summary(mod)

Call:
lm(formula = skullW ~ headL, data = possum)

Residuals:
    Min      1Q  Median      3Q     Max
-4.6263 -1.0783 -0.1128  0.6412 11.3465

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.46871    5.62328  -0.083    0.934
headL        0.61934    0.06068  10.207   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.201 on 102 degrees of freedom
Multiple R-squared:  0.5053,    Adjusted R-squared:  0.5004
F-statistic: 104.2 on 1 and 102 DF,  p-value: < 2.2e-16

>
> class(mod)
[1] "lm"
>
> typeof(mod)
[1] "list"
```



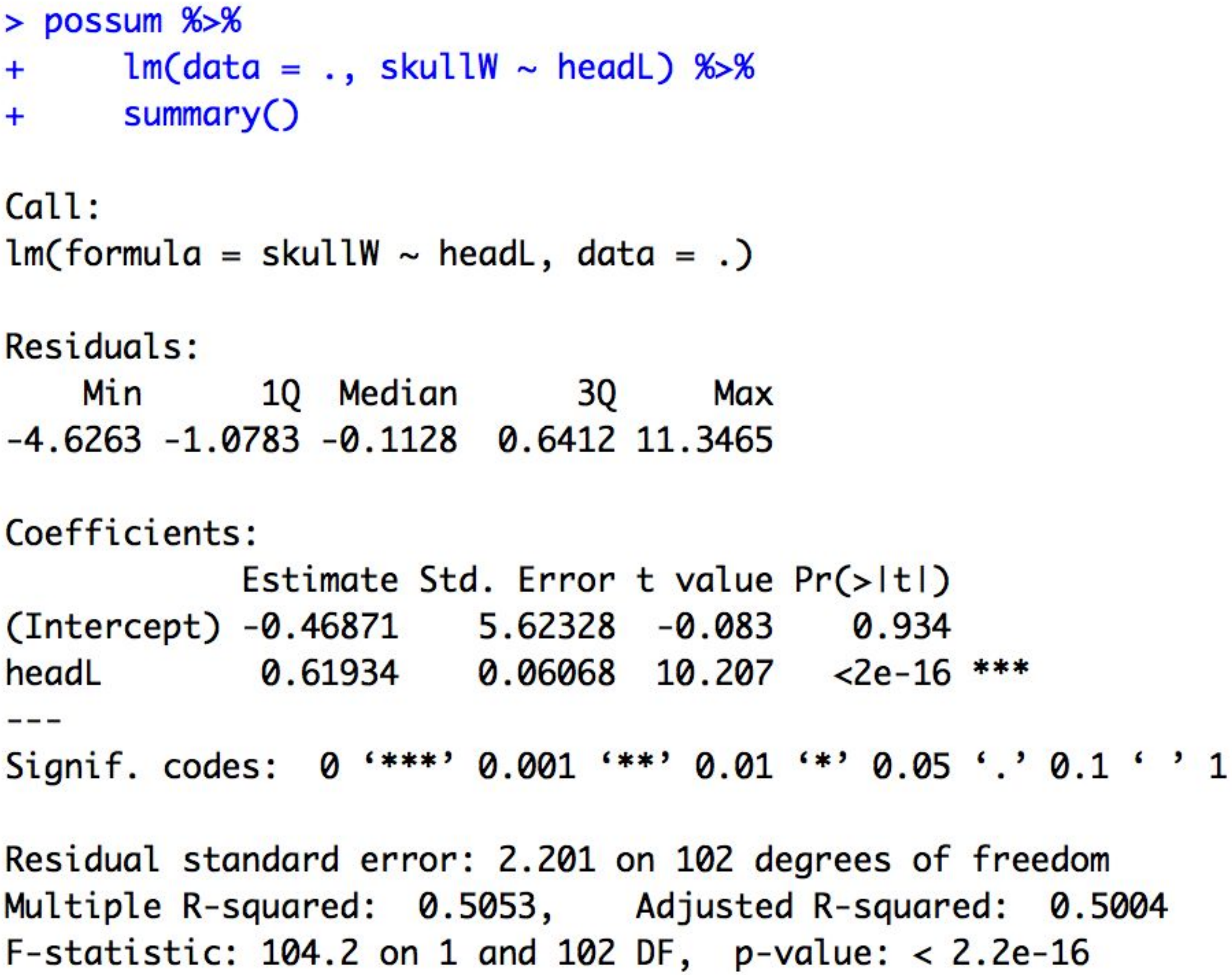
```
> str(mod)
List of 12
 $ coefficients : Named num [1:2] -0.469 0.619
  ..- attr(*, "names")= chr [1:2] "(Intercept)" "headL"
 $ residuals    : Named num [1:104] 2.5891 0.7801 2.2511 -0.1535 0.0994 ...
  ..- attr(*, "names")= chr [1:104] "1" "2" "3" "4" ...
 $ effects      : Named num [1:104] -580.102 22.461 2.039 -0.378 -0.15 ...
  ..- attr(*, "names")= chr [1:104] "(Intercept)" "headL" "" "" ...
 $ rank         : int 2
 $ fitted.values: Named num [1:104] 57.8 56.8 57.7 57.3 56.2 ...
  ..- attr(*, "names")= chr [1:104] "1" "2" "3" "4" ...
 $ assign       : int [1:2] 0 1
 $ qr           :List of 5
  ..$ qr   : num [1:104, 1:2] -10.198 0.0981 0.0981 0.0981 0.0981 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:104] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:2] "(Intercept)" "headL"
  .. ..- attr(*, "assign")= int [1:2] 0 1
  ..$ qraux: num [1:2] 1.1 1.01
  ..$ pivot: int [1:2] 1 2
  ..$ tol  : num 1e-07
  ..$ rank : int 2
  ..- attr(*, "class")= chr "qr"
 $ df.residual  : int 102
 $ xlevels      : Named list()
 $ call         : language lm(formula = skullW ~ headL, data = possum)
 $ terms        :Classes 'terms', 'formula'  language skullW ~ headL
  .. ..- attr(*, "variables")= language list(skullW, headL)
  .. ..- attr(*, "factors")= int [1:2, 1] 0 1
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..$ : chr [1:2] "skullW" "headL"
  .. .. .. ..$ : chr "headL"
  .. ..- attr(*, "term.labels")= chr "headL"
  .. ..- attr(*, "order")= int 1
  .. ..- attr(*, "intercept")= int 1
  .. ..- attr(*, "response")= int 1
  .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
  .. ..- attr(*, "predvars")= language list(skullW, headL)
  .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
  .. .. ..- attr(*, "names")= chr [1:2] "skullW" "headL"
 $ model        :'data.frame':  104 obs. of  2 variables:
  ..$ skullW: num [1:104] 60.4 57.6 60 57.1 56.3 54.8 58.2 57.6 56.3 58 ...
  ..$ headL : num [1:104] 94.1 92.5 94 93.2 91.5 93.1 95.3 94.8 93.4 91.8 ...
  ..- attr(*, "terms")=Classes 'terms', 'formula'  language skullW ~ headL
  .. .. ..- attr(*, "variables")= language list(skullW, headL)
  .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
  .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. ..$ : chr [1:2] "skullW" "headL"
  .. .. .. .. ..$ : chr "headL"
  .. .. ..- attr(*, "term.labels")= chr "headL"
  .. .. ..- attr(*, "order")= int 1
  .. .. ..- attr(*, "intercept")= int 1
  .. .. ..- attr(*, "response")= int 1
```

Linear Regression

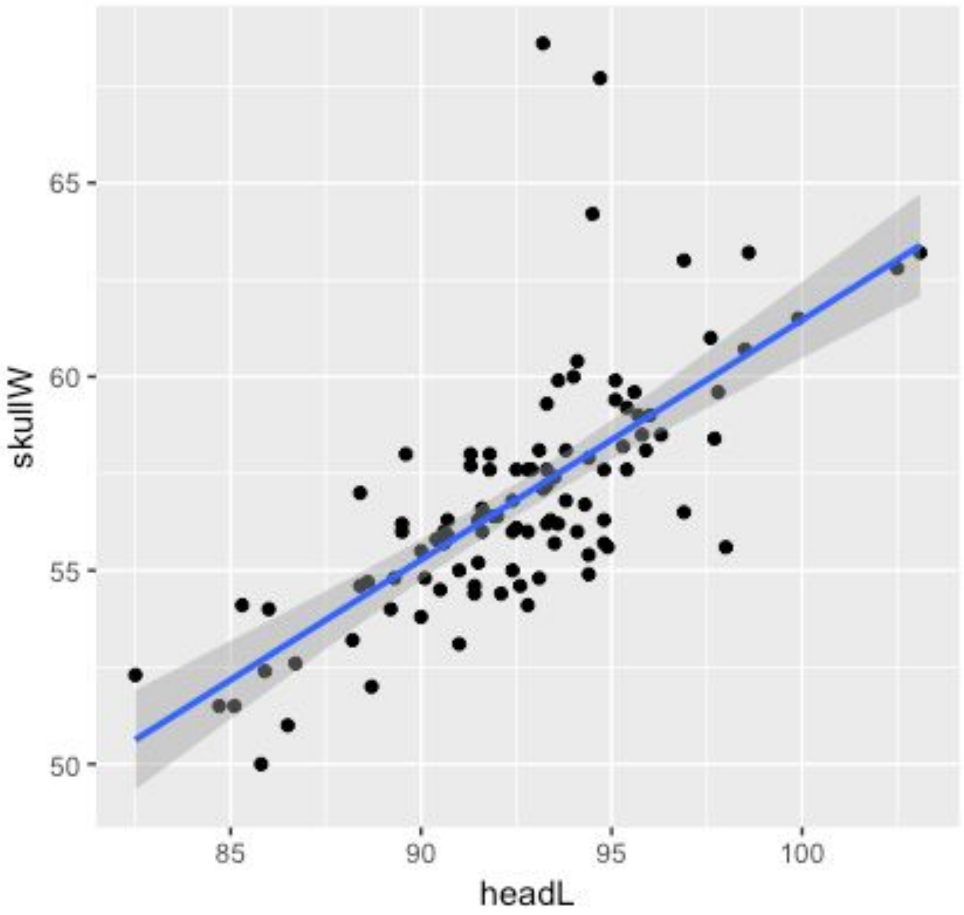


```
> str(summary(mod))
List of 11
 $ call         : language lm(formula = skullW ~ headL, data = possum)
 $ terms        :Classes 'terms', 'formula'  language skullW ~ headL
  .. ..- attr(*, "variables")= language list(skullW, headL)
  .. ..- attr(*, "factors")= int [1:2, 1] 0 1
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..$ : chr [1:2] "skullW" "headL"
  .. .. .. ..$ : chr "headL"
  .. ..- attr(*, "term.labels")= chr "headL"
  .. ..- attr(*, "order")= int 1
  .. ..- attr(*, "intercept")= int 1
  .. ..- attr(*, "response")= int 1
  .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
  .. ..- attr(*, "predvars")= language list(skullW, headL)
  .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
  .. .. ..- attr(*, "names")= chr [1:2] "skullW" "headL"
 $ residuals    : Named num [1:104] 2.5891 0.7801 2.2511 -0.1535 0.0994 ...
  ..- attr(*, "names")= chr [1:104] "1" "2" "3" "4" ...
 $ coefficients : num [1:2, 1:4] -0.4687 0.6193 5.6233 0.0607 -0.0834 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:2] "(Intercept)" "headL"
  .. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"
 $ aliased      : Named logi [1:2] FALSE FALSE
  ..- attr(*, "names")= chr [1:2] "(Intercept)" "headL"
 $ sigma        : num 2.2
 $ df           : int [1:3] 2 102 2
 $ r.squared    : num 0.505
 $ adj.r.squared: num 0.5
 $ fstatistic   : Named num [1:3] 104 1 102
  ..- attr(*, "names")= chr [1:3] "value" "numdf" "dendf"
 $ cov.unscaled : num [1:2, 1:2] 6.52981 -0.07041 -0.07041 0.00076
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:2] "(Intercept)" "headL"
  .. ..$ : chr [1:2] "(Intercept)" "headL"
 - attr(*, "class")= chr "summary.lm"
```

Linear Regression



```
> possum %>%
+     lm(data = ., skullW ~ headL) %>%
+     summary()

Call:
lm(formula = skullW ~ headL, data = .)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.6263 -1.0783 -0.1128  0.6412 11.3465 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.46871    5.62328  -0.083    0.934    
headL        0.61934    0.06068  10.207   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.201 on 102 degrees of freedom
Multiple R-squared:  0.5053,	Adjusted R-squared:  0.5004 
F-statistic: 104.2 on 1 and 102 DF,  p-value: < 2.2e-16
```

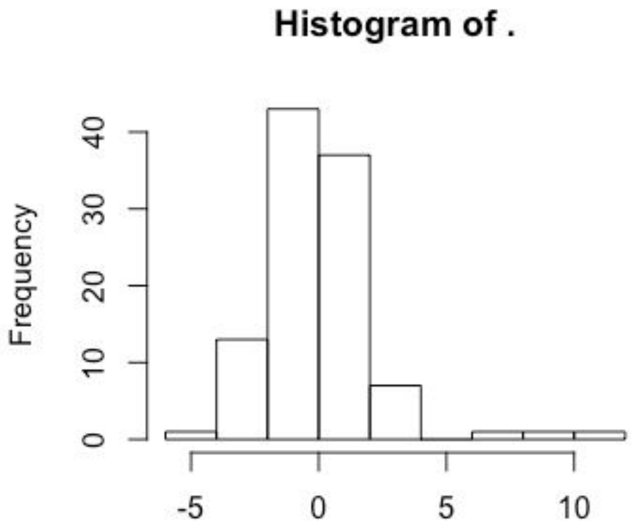


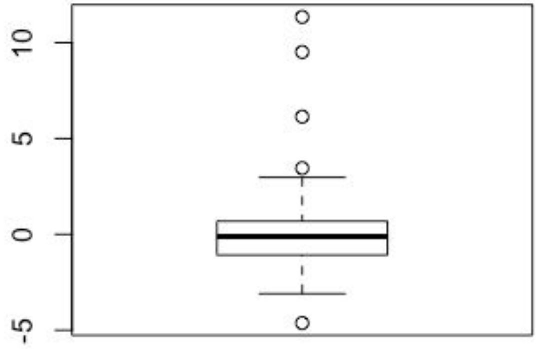
Linear Regression

```
> coefficients(mod)
(Intercept)       headL
 -0.4687115   0.6193367
```

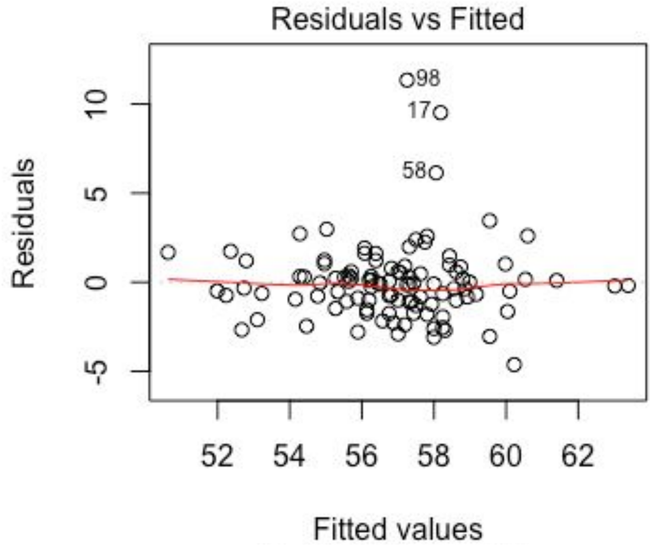
```
> fitted.values(mod) %>% head()
       1        2        3        4        5        6
57.81087 56.81993 57.74894 57.25347 56.20060 57.19154
> residuals(mod) %>% head()
          1           2           3           4           5           6
 2.58912765  0.78006637  2.25106132 -0.15346932  0.09940308 -2.39153565
```

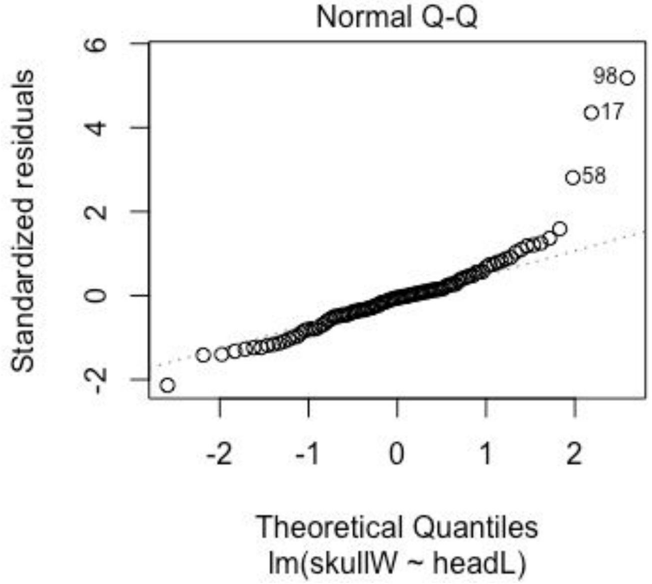
```
> residuals(mod) %>% hist()
> residuals(mod) %>% boxplot()
```



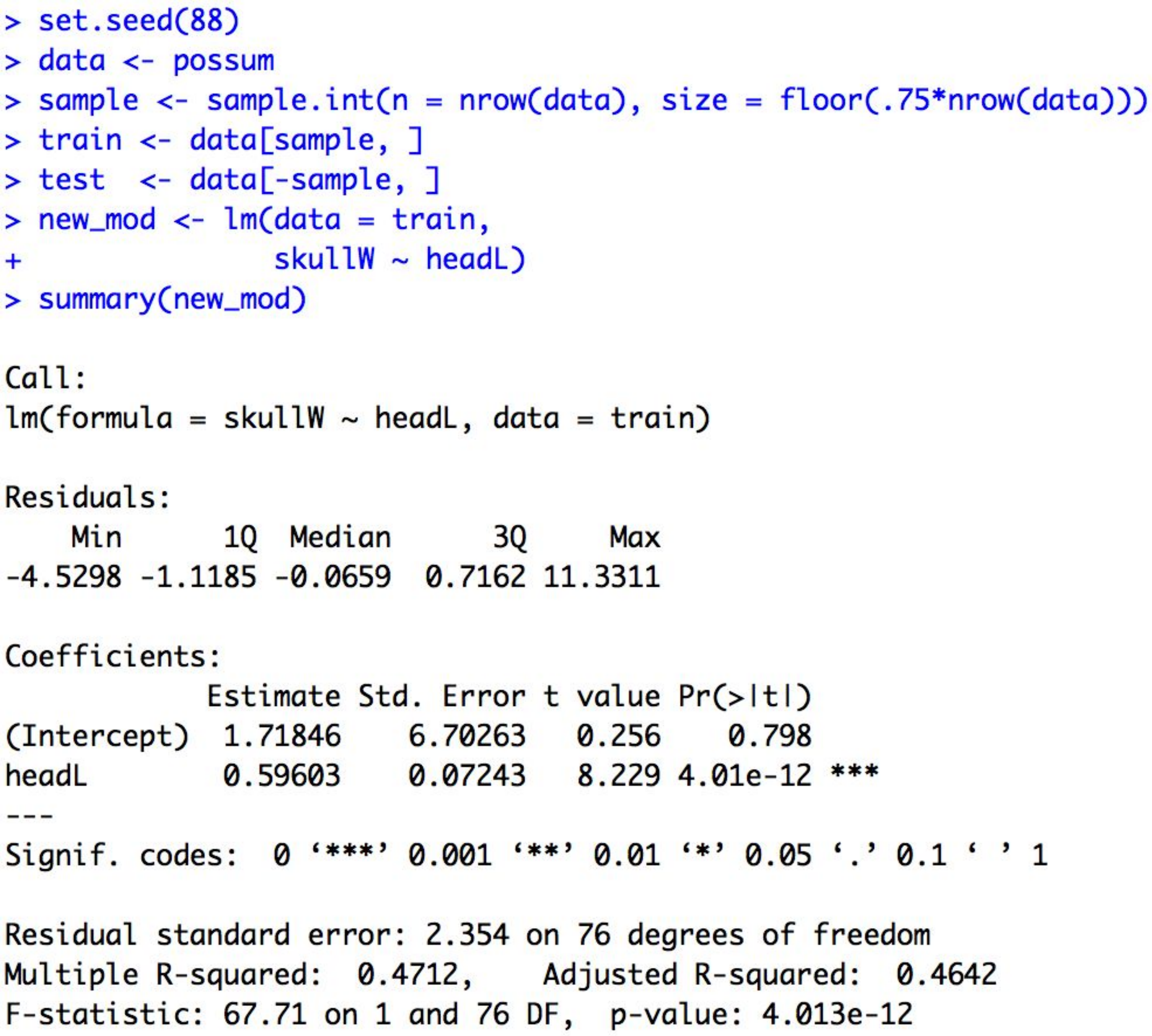


```
> plot(mod, which = c(1,2))
```

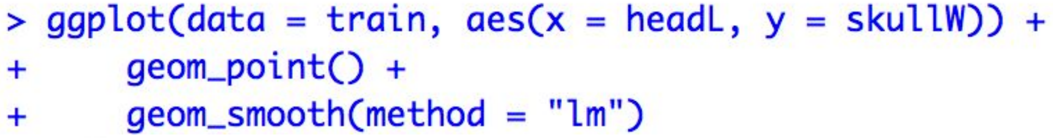




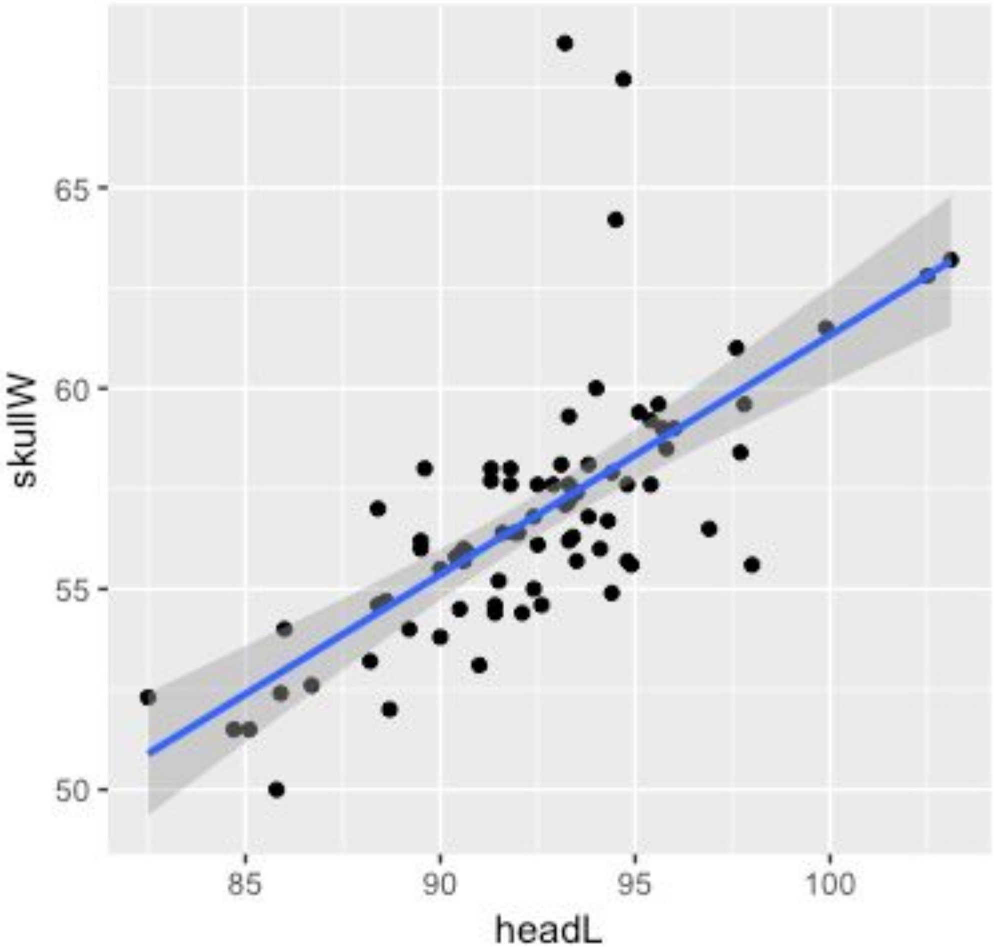
Linear Regression



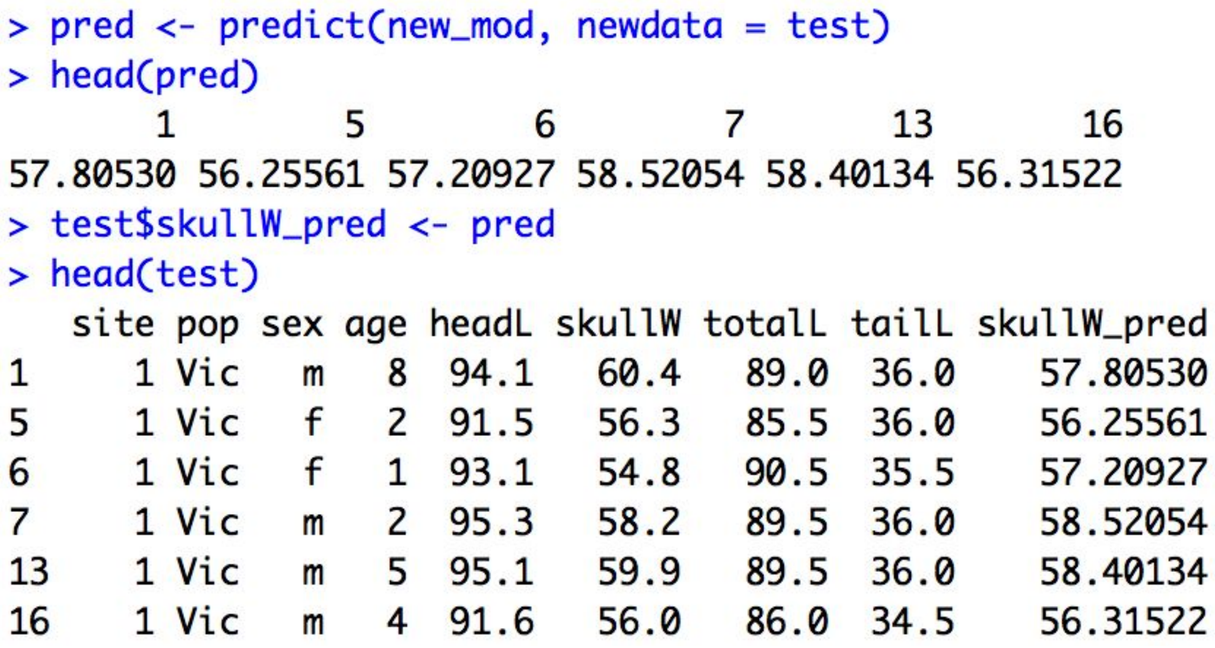
```
> set.seed(88)
> data <- possum
> sample <- sample.int(n = nrow(data), size = floor(.75*nrow(data)))
> train <- data[sample, ]
> test  <- data[-sample, ]
> new_mod <- lm(data = train,
+               skullW ~ headL)
> summary(new_mod)

Call:
lm(formula = skullW ~ headL, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.5298 -1.1185 -0.0659  0.7162 11.3311 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.71846    6.70263   0.256    0.798    
headL        0.59603    0.07243   8.229 4.01e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.354 on 76 degrees of freedom
Multiple R-squared:  0.4712,	Adjusted R-squared:  0.4642 
F-statistic: 67.71 on 1 and 76 DF,  p-value: 4.013e-12
```



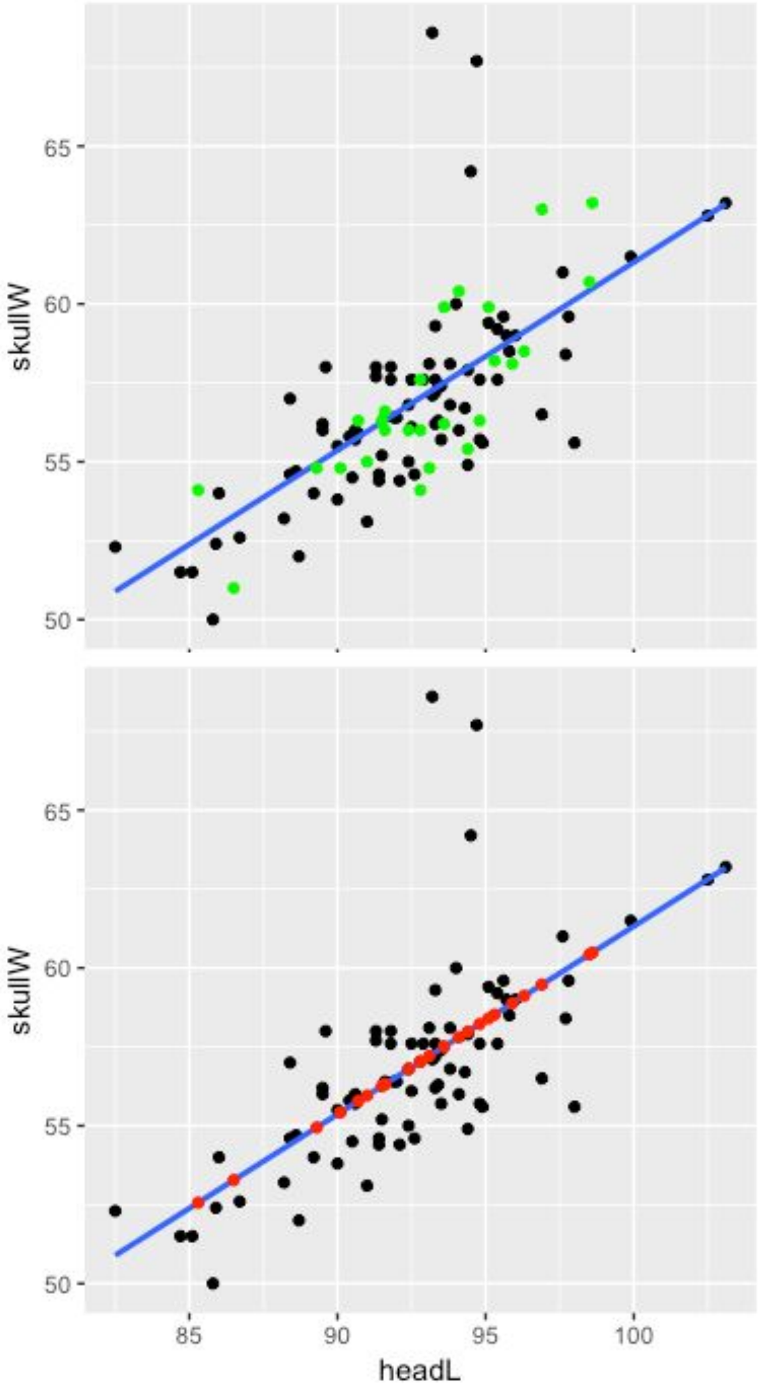
```
> ggplot(data = train, aes(x = headL, y = skullW)) +
+     geom_point() +
+     geom_smooth(method = "lm")
```

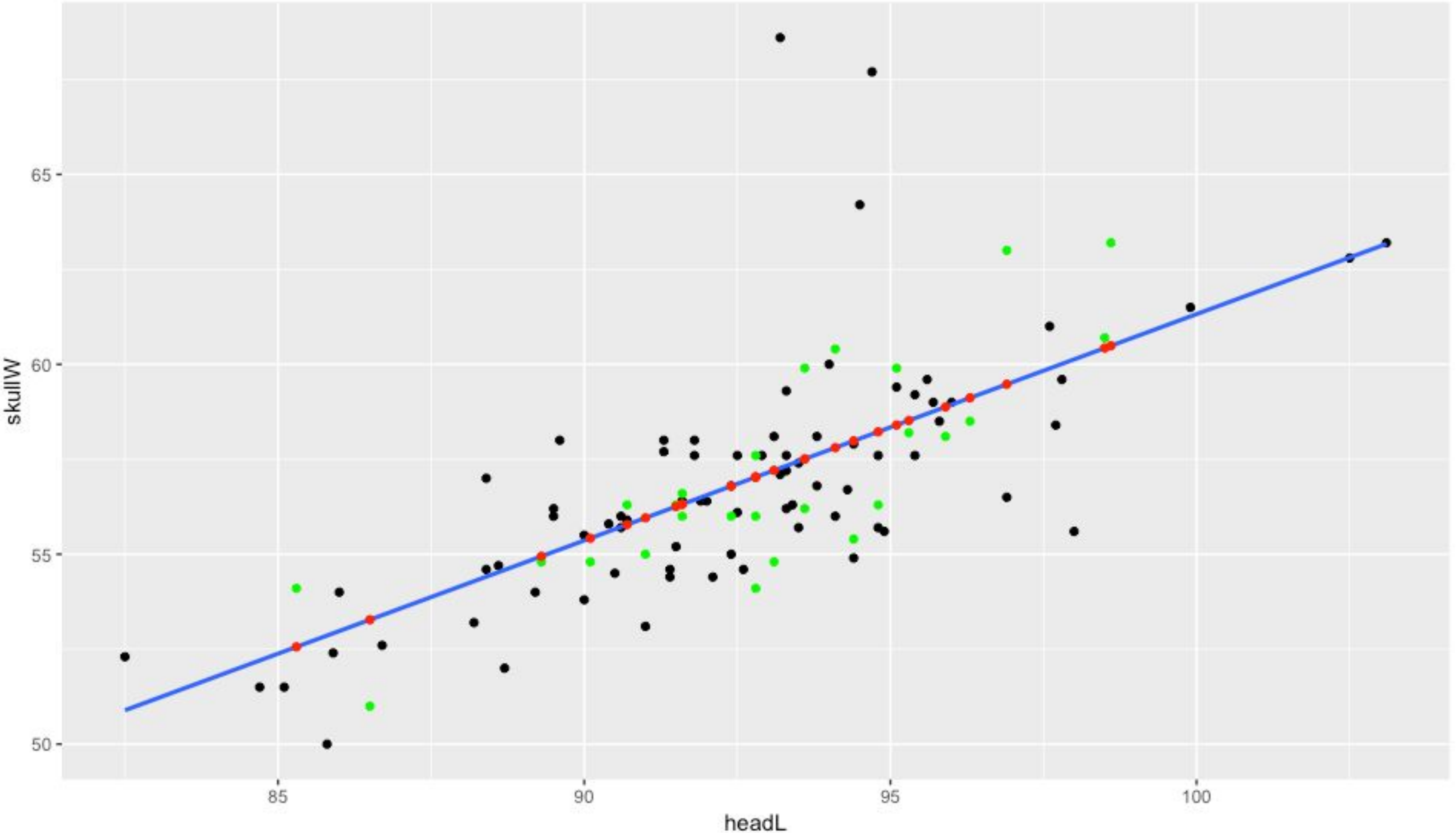


Linear Regression



```
> pred <- predict(new_mod, newdata = test)
> head(pred)
       1        5        6        7       13       16 
57.80530 56.25561 57.20927 58.52054 58.40134 56.31522 
> test$skullW_pred <- pred
> head(test)
   site pop sex age headL skullW totalL tailL skullW_pred
1     1 Vic   m   8  94.1   60.4   89.0  36.0    57.80530
5     1 Vic   f   2  91.5   56.3   85.5  36.0    56.25561
6     1 Vic   f   1  93.1   54.8   90.5  35.5    57.20927
7     1 Vic   m   2  95.3   58.2   89.5  36.0    58.52054
13    1 Vic   m   5  95.1   59.9   89.5  36.0    58.40134
16    1 Vic   m   4  91.6   56.0   86.0  34.5    56.31522
```




skullW
65
60
55
50
85
90
95
100
headL

Tasks 5 and 6 – Case Study

Anscombe's data set

- Scatter plot facetted by set
- Summary calculation (mean, sd) grouped by set
- Pearson's correlation by set, and non-parametric, and p-values
- Add `geom_smooth()` to the plot

Other data set: https://archive.ics.uci.edu/ml/datasets/Air+quality

- Explore data set, clean if needed
- Explore each variable independently
- Cross correlations
- Build simple linear models with each predictor, check assumptions
- For one of the models create train-test sets, plot the model, for the test set color real and predicted points differently; R^2 and p-value to title

Task – Case Study

- Useful functions:
    - `duplicated()`
    - `sum(), prod()`
    - `which()`
    - `pairs()`
    - `cor()`
    - `corrplot::corrplot()`
    - `corrplot::cor.mtest()`