STATS 3001 / STATS 4104 / STATS 7054
Statistical Modelling III
Practical 1 - Linear model recap
Solutions

Week 1

The following exercises are intended as revision of basic regression calculation and matrix manipulation functions in R.

For the purpose of this exercise, we will use built-in data set `Rubber` that is provided with the `MASS` library. The data set comprises three variables recorded on thirty samples of tyre rubber that were being tested for durability:

- `loss`: The abrasion loss in grams per hour;
- `hard`: The hardness in Shore units; and
- `tens`: The tensile strength in kg per square metre.

STEPS

- Load packages using the command

```
pacman::p_load(tidyverse, gglm)
```

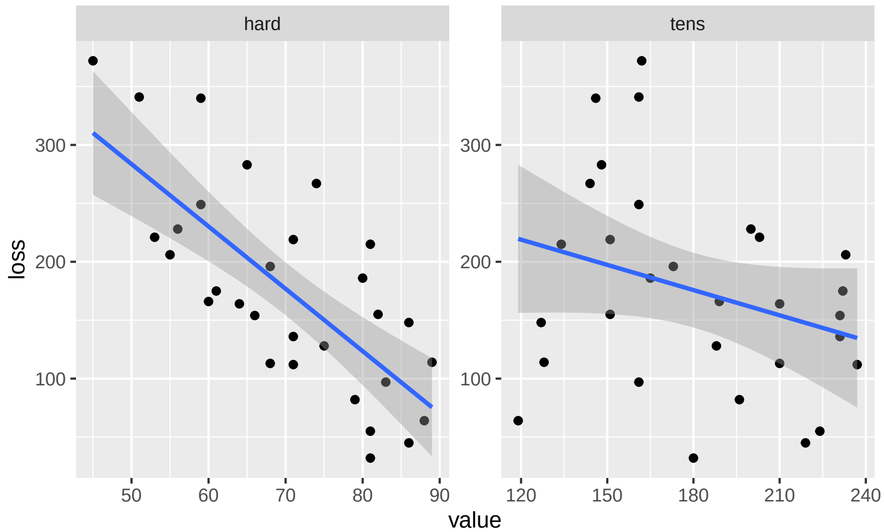
- Load the `rubber` dataset from the `MASS` package

```
data(Rubber, package = "MASS")
```

- Obtain scatter-plots of loss against each of the other predictors.

```
Rubber %>%
  pivot_longer(-loss) %>%
  ggplot(aes(value, loss)) +
  geom_point() +
  facet_wrap(~name, scales = "free") +
  geom_smooth(method = lm)
```

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



- Use the `lm()` function to fit the following model.

$$E(\texttt{loss}_i) = \beta_0 + \beta_1 \times \texttt{hard}_i + \beta_2 \times \texttt{tens}_i.$$

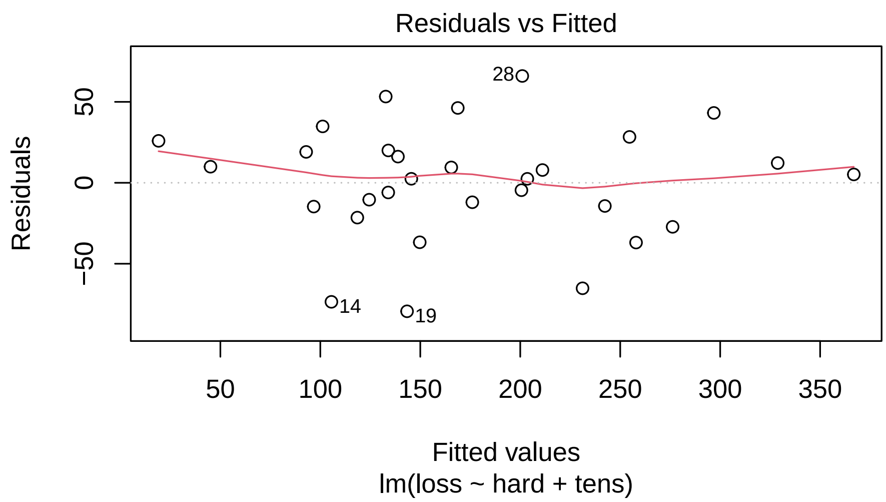
```
rubber_lm <- lm(loss ~ hard + tens, data = Rubber)
summary(rubber_lm)
```

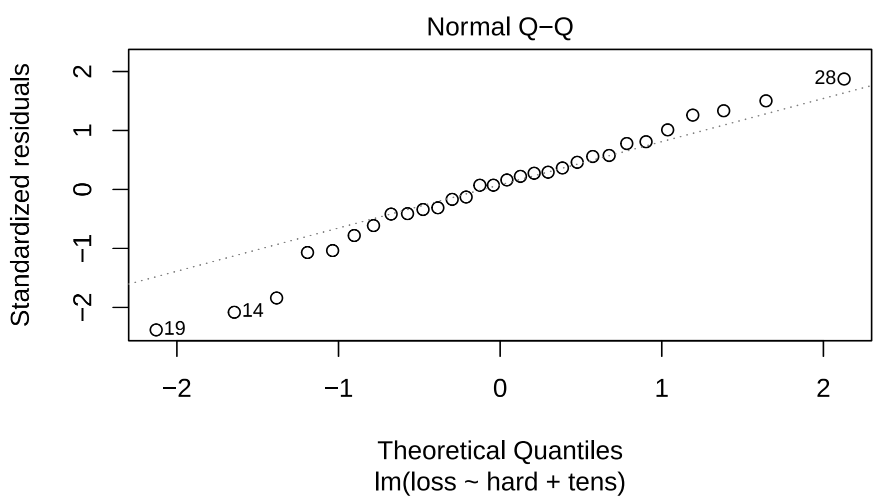
```
##
## Call:
## lm(formula = loss ~ hard + tens, data = Rubber)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.385 -14.608   3.816  19.755  65.981
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 885.1611    61.7516  14.334 3.84e-14 ***
## hard         -6.5708     0.5832 -11.267 1.03e-11 ***
## tens         -1.3743     0.1943  -7.073 1.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.49 on 27 degrees of freedom
## Multiple R-squared:  0.8402, Adjusted R-squared:  0.8284
## F-statistic:    71 on 2 and 27 DF,  p-value: 1.767e-11
```

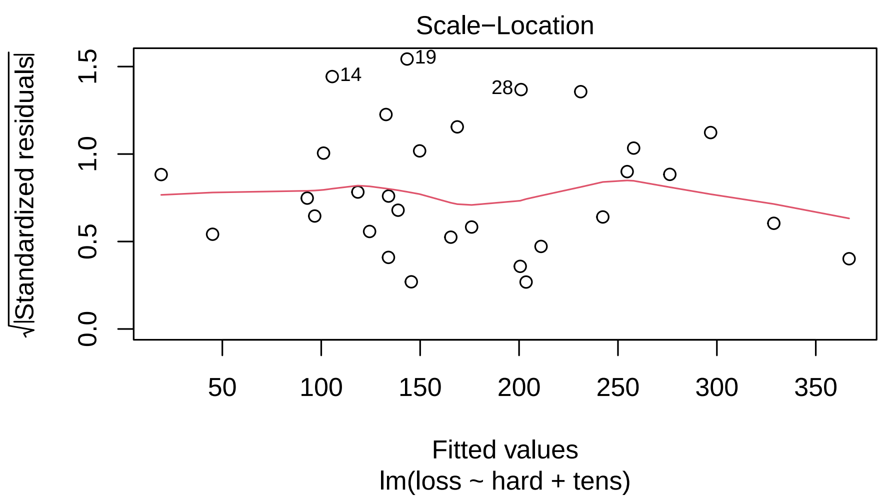
- Interpret the output
- The least squares estimate of β_1 is $\hat{\beta}_1 = -6.57$. The interpretation of this estimate is that and increase of 1 unit of hardness, with tensile strength kept constant, would reduce the rate of abrasion by 6.57 grams per hour.
- The least squares estimate of β_2 is $\hat{\beta}_2 = -1.37$. The interpretation of this estimate is that and increase of 1 unit of tensile strength, with hardness kept constant, would reduce the rate of abrasion by 1.37 grams per hour.

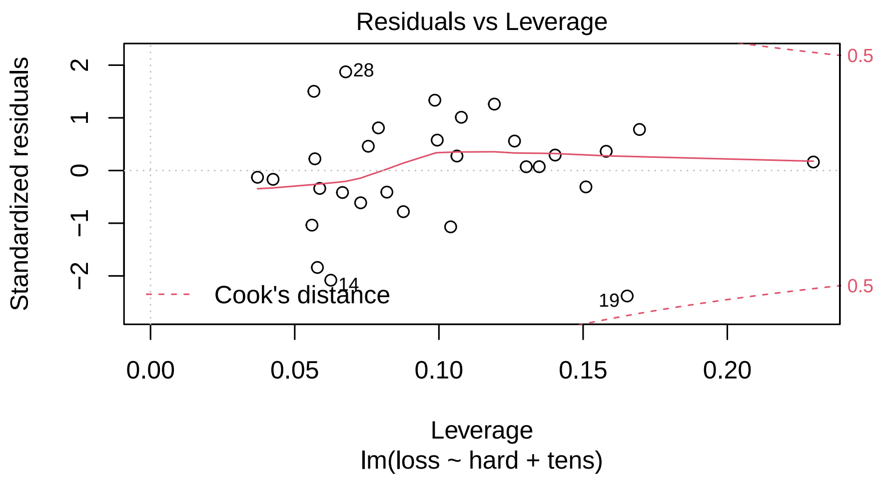
- What are the assumptions of the linear regression model. Produce appropriate plots to check them.

```
plot(rubber_lm)
```

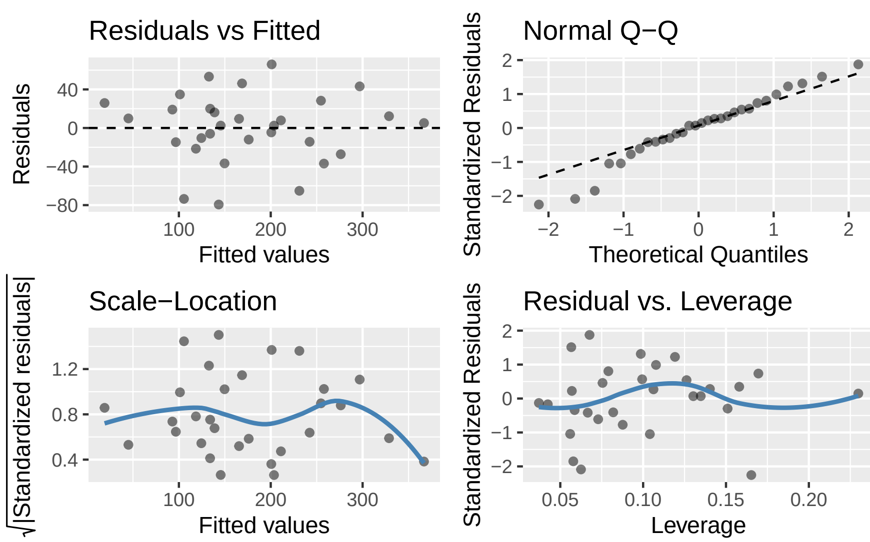









```
gglm(rubber_lm)
```



Linearity:

The residuals are roughly randomly scattered about the zero line in the residuals versus fitted values plot, apart from slight curvature near the endpoints. The residuals versus hardness plot shows a random scatter but the slight curvature is apparent in the residuals versus tensile strength plot. On balance, the linearity assumption is close to reasonable.

Homoscedasticity

The spread about the zero line appears roughly constant in all three plots indicating that the assumption of constant variance is reasonable.

Normality

There is some departure from normality in the lower tail of the distribution of residuals, with more large negative residuals than expected for a normal distribution. The bulk of the data is close to normally distributed however.

- Predict the loss for the following points:

hard	tens
50	200
65	190

- Calculate 95% CI and PI for the above points.

```r
new_pts <- tibble(
  hard = c(50, 65),
  tens = c(200, 190)
)
new_pts
```

```
## # A tibble: 2 x 2
##    hard  tens
##   <dbl> <dbl>
## 1    50   200
## 2    65   190
```

```r
predict(rubber_lm)
```

```
##         1         2         3         4         5         6         7         8
## 366.83526 203.55082 165.50016 134.02032 101.16617  92.92030  45.07805  19.09546
##         9        10        11        12        13        14        15        16
## 257.92184 231.16639 176.02253 149.73921  96.70044 105.54777 242.33228 200.58874
##        17        18        19        20        21        22        23        24
## 133.97826 118.51804 143.38498 276.21795 211.11111 132.73328 138.83198 124.44535
##        25        26        27        28        29        30
## 328.78459 296.83263 254.65903 201.01880 168.76611 145.53215
```

```r
predict(rubber_lm, newdata = new_pts)
```

```
##        1        2
## 281.7573 196.9379
```

```r
predict(rubber_lm, newdata = new_pts, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 281.7573 254.8762 308.6383
## 2 196.9379 181.8821 211.9938
```

```r
predict(rubber_lm, newdata = new_pts, interval = "prediction")
```

```
##        fit      lwr      upr
## 1 281.7573 202.2079 361.3066
## 2 196.9379 120.5692 273.3067
```

- Get the design matrix of the model using the command `model.matrix()`.

```r
X <- model.matrix(rubber_lm)
X
```

```
##    (Intercept) hard tens
## 1            1   45  162
## 2            1   55  233
## 3            1   61  232
## 4            1   66  231
## 5            1   71  231
## 6            1   71  237
## 7            1   81  224
## 8            1   86  219
## 9            1   53  203
## 10           1   60  189
## 11           1   64  210
## 12           1   68  210
## 13           1   79  196
## 14           1   81  180
## 15           1   56  200
## 16           1   68  173
## 17           1   75  188
## 18           1   83  161
## 19           1   88  119
## 20           1   59  161
## 21           1   71  151
## 22           1   80  165
## 23           1   82  151
## 24           1   89  128
## 25           1   51  161
## 26           1   59  146
## 27           1   65  148
## 28           1   74  144
## 29           1   81  134
## 30           1   86  127
## attr(,"assign")
## [1] 0 1 2
```

- Assign the response variable `loss` to a variable `Y`.

```r
Y <- Rubber$loss
Y
```

```
##  [1] 372 206 175 154 136 112  55  45 221 166 164 113  82  32 228 196 128  97  64
## [20] 249 219 186 155 114 341 340 283 267 215 148
```

- Using the R commands `%*%`, `solve()`, and `t()`, calculate

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

```r
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% Y
beta_hat
```

```
##                    [,1]
## (Intercept) 885.161109
## hard         -6.570830
## tens         -1.374312
```

```
summary(rubber_lm)
```

```
##
## Call:
## lm(formula = loss ~ hard + tens, data = Rubber)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.385 -14.608   3.816  19.755  65.981
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 885.1611    61.7516  14.334 3.84e-14 ***
## hard         -6.5708     0.5832 -11.267 1.03e-11 ***
## tens         -1.3743     0.1943  -7.073 1.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.49 on 27 degrees of freedom
## Multiple R-squared:  0.8402, Adjusted R-squared:  0.8284
## F-statistic:    71 on 2 and 27 DF,  p-value: 1.767e-11
```

Is it the same as the answer from `lm()`?

- Calculate the fitted values

$$\hat{\boldsymbol{\eta}} = X\hat{\boldsymbol{\beta}}$$

```
eta <- X %*% beta_hat
head(eta)
```

```
##       [,1]
## 1 366.8353
## 2 203.5508
## 3 165.5002
## 4 134.0203
## 5 101.1662
## 6  92.9203
```

- Calculate the residual variance, s_e^2 directly from the observed and fitted values. Compare the result to the `residual standard error` produced by `lm()`

```
n <- nrow(Rubber)
se <- sqrt(sum((Y - eta)^2) / (n - 3))
se
```

```
## [1] 36.48934
```

- Calculate the estimated variance matrix for $\hat{\boldsymbol{\beta}}$ using

$$(X^T X)^{-1} \times s_e^2$$

  Compare this to the result of the built-in calculation `vcov()`.

```
vcov(rubber_lm)
```

```
##             (Intercept)         hard        tens
## (Intercept)  3813.25821 -30.01766345 -9.19635018
## hard          -30.01766   0.34010794  0.03390882
## tens           -9.19635   0.03390882  0.03775595
```

```
solve(t(X) %*% X) * se^2
```

```
##             (Intercept)         hard        tens
## (Intercept)  3813.25821 -30.01766345 -9.19635018
## hard          -30.01766   0.34010794  0.03390882
## tens           -9.19635   0.03390882  0.03775595
```