Introduction to Statistics with R

Session R03: Regression

Marvin Schmitt

The example data set

We analyze structures in the **Chicago Face Database** (Ma et al., 2015). Each row refers to a portrait which was rated with respect to different categories by a sample of raters.

```
df = read.csv("R03_notes_dataset.csv")
nrow(df)
```

```
## [1] 597
```

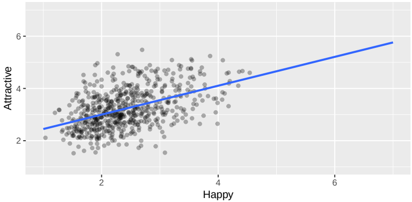
```
colnames(df)
```

```
##  [1] "ID"            "Gender"        "Age"           "Afraid"
##  [5] "Angry"         "Attractive"    "Babyface"      "Disgusted"
##  [9] "Dominant"      "Feminine"      "Happy"         "Masculine"
## [13] "Prototypic"    "Sad"           "Surprised"     "Threatening"
## [17] "Trustworthy"   "Unusual"       "Nose_Width"    "Nose_Length"
## [21] "FaceRoundness" "Noseshape"
```

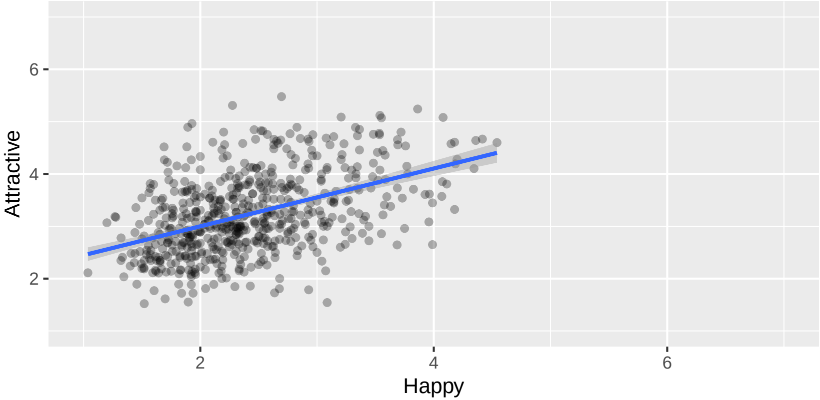
ggplot2: Scatterplots with fitted line

We can fit (linear) regression curves in `ggplot` scatterplots with `geom_smooth()`.

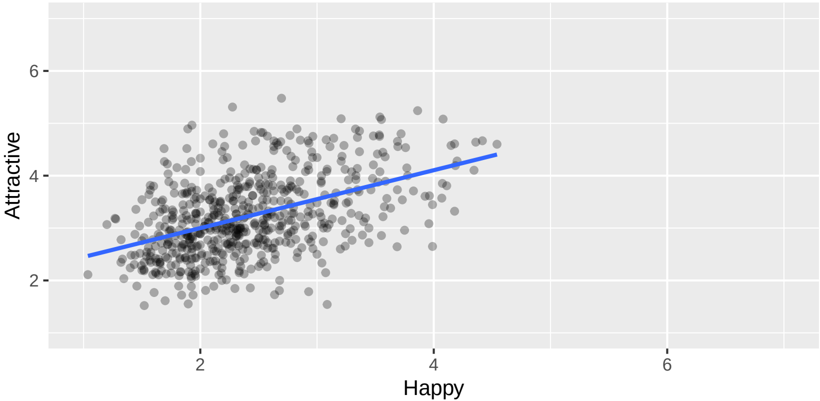
- `method='lm'` uses a **linear model**
- `se=FALSE` does not print the confidence interval around the fitted line
- `fullrange=TRUE` continues the line beyond the data range



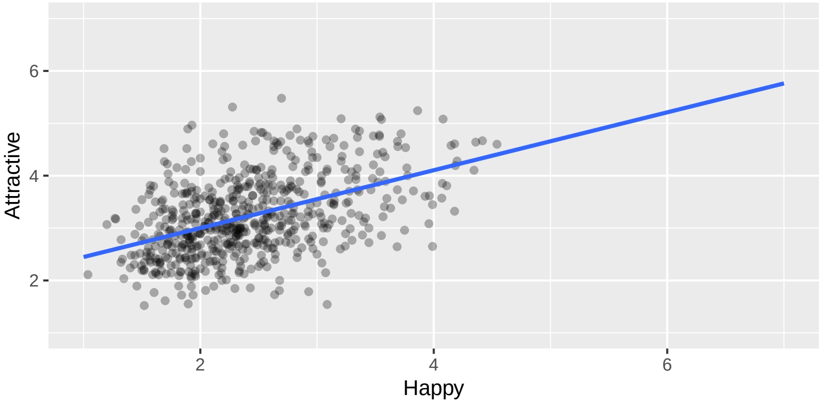
```
ggplot(df, aes(x=Happy, y=Attractive)) + xlim(1,7) + ylim(1,7) +
  geom_point(alpha=.30) +
  geom_smooth(method="lm")
```



```
ggplot(df, aes(x=Happy, y=Attractive)) + xlim(1,7) + ylim(1,7) +
  geom_point(alpha=.30) +
  geom_smooth(method="lm", se=FALSE)
```



```r
ggplot(df, aes(x=Happy, y=Attractive)) + xlim(1,7) + ylim(1,7) +
  geom_point(alpha=.30) +
  geom_smooth(method="lm", se=FALSE, fullrange=TRUE)
```



Again, we can fit separate regression lines based on a grouping variable, i.e. Gender:

```
ggplot(df, aes(x=Masculine, y=Attractive, color=Gender))+xlim(1,7)+ylim(1,7)+
  geom_point(alpha=.30) +
  stat_cor(p.accuracy = 0.01) +
  geom_smooth(method="lm")
```

Calculating multiple regression numerically: `lm`

The `lm()` function fits a **linear model** to the data. It uses the formula syntax `y ~ x_1 + x_2 + ... + x_k`.

```
model = lm(Attractive ~ Feminine + Happy + Afraid, data=df)
summary(model)
```

```
##
## Call:
## lm(formula = Attractive ~ Feminine + Happy + Afraid, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.60554 -0.41396 -0.01182  0.40463  1.70398
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.69062    0.18882   8.953   <2e-16 ***
## Feminine     0.22157    0.01621  13.667   <2e-16 ***
## Happy        0.43758    0.04140  10.570   <2e-16 ***
## Afraid      -0.10779    0.06727  -1.602     0.11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5958 on 593 degrees of freedom
## Multiple R-squared:  0.4023, Adjusted R-squared:  0.3993
## F-statistic: 133.1 on 3 and 593 DF,  p-value: < 2.2e-16
```

Testing assumptions

In order to yield interpretable results, the data for a regression analysis should fulfil the following criteria:

- No multivariate outliers
- No multicollinearity or singularity
- Homoscedasticity (w.r.t. the residuals)
- Normal residuals
- Linear relation between predictor(s) and criterion

We save the linear model as an object:

```r
model = lm(Attractive ~ Feminine + Happy + Afraid, data=df)
```

Multivariate outliers

Values with standardized residuals > 3 or < -3 are possible outliers (James et al., 2014).

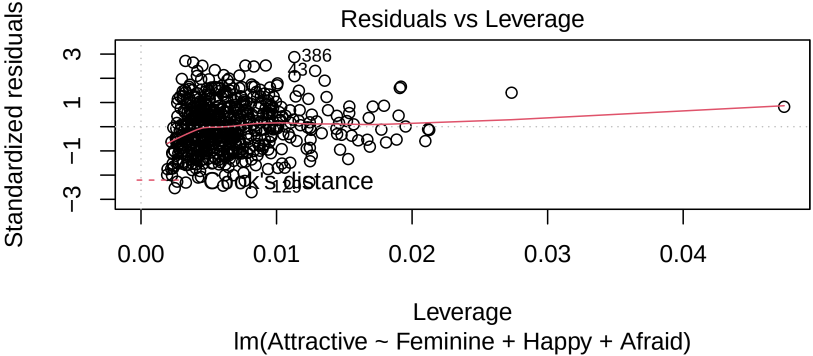
The **leverage statistic** quantifies if points have extreme predictor values. A leverage above $\frac{2(k+1)}{N}$ indicates observations with high leverage (Bruce & Bruce, 2017). k is the number of predictors, N is the sample size.

```
k = length(model$coefficients) - 1 # number of predictors = 3
N = nobs(model)    # number of observations = 597
critical_leverage = (2 * (k + 1)) / N

print(critical_leverage)
```

```
## [1] 0.01340034
```

```r
plot(model, 5)
```



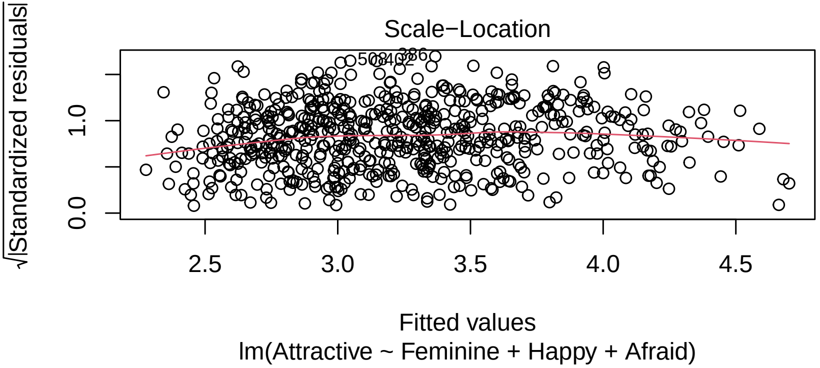
Multicollinearity

```r
df %>%
  select(Attractive, Feminine, Happy, Afraid) %>%
  cor() %>%
  round(2)
```

```
##            Attractive Feminine Happy Afraid
## Attractive       1.00     0.50  0.46  -0.16
## Feminine         0.50     1.00  0.17   0.06
## Happy            0.46     0.17  1.00  -0.35
## Afraid          -0.16     0.06 -0.35   1.00
```

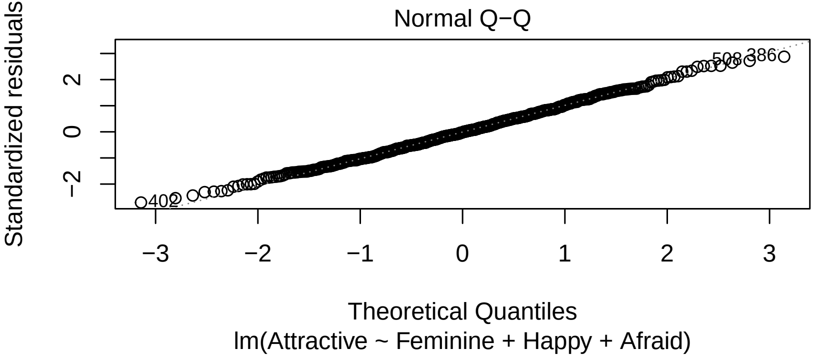
Homoscedasticity

```r
plot(model, 3)
```



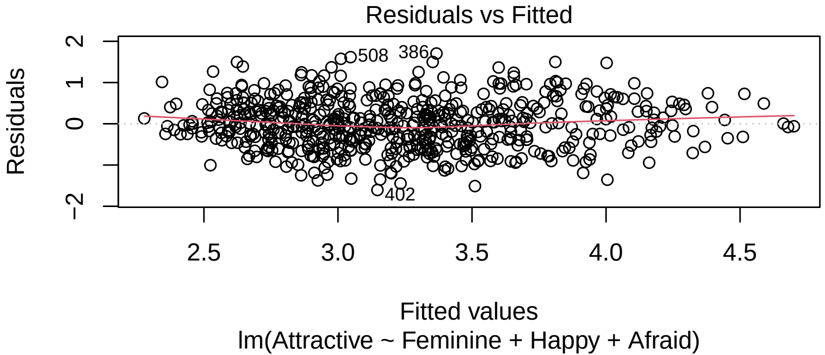
Normal residuals

```
plot(model, 2)
```



Linearity

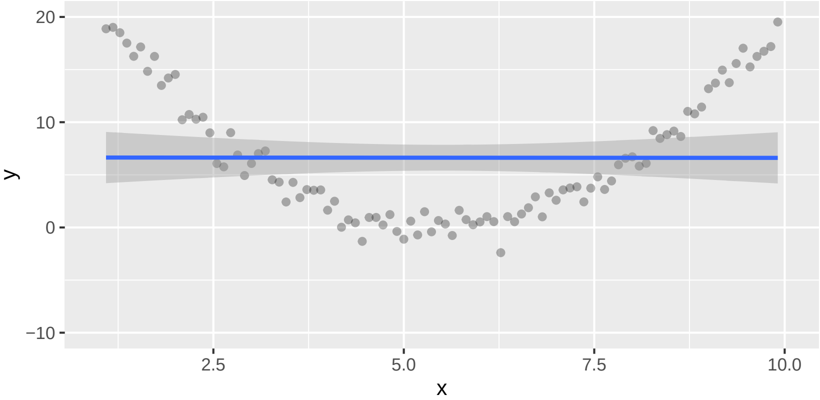
```
plot(model, 1)
```



Addressing the correlation's 'other caveat':

```
ggplot(df_xy, aes(x=x, y=y)) + geom_point(alpha=.30) + ylim(-10, 20) +
geom_smooth(method="lm")
```

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



```r
ggplot(df_xy, aes(x=x, y=y)) + geom_point(alpha=.30) + ylim(-10, 20) +
geom_smooth(method="lm", formula="y~x+I(x^2)")
```