Supplement 2

Oxygen saturation (SaO2)

Peter Kamerman and Etienne Basson

Last updated: 23 May 2020

Contents

1 Research question

1. Determine whether there is an association between blood colour components (CIE L*a*b*) and saturation of haemoglobin.

2. Generate a predictive model of the (CIE L*a*b*) colour component from saturation of haemoglobin.

---

2 Load data

```
data <- read_rds('data-cleaned/clean-data.rds')
```

---

3 Quick look at the data

```
dim(data)
```

```
## [1] 163  22
```

```
names(data)
```

```
##  [1] "Animal_ID"        "Trial"            "Dpat"
##  [4] "Time_min"         "l_star"           "a_star"
##  [7] "b_star"           "PaO2"             "cal_SaO2"
## [10] "PaCO2"            "Cl"               "HCO3"
## [13] "Temp"             "pH"               "Cartridge_expired"
## [16] "Coox_SaO2"        "Hct"              "total_Hb"
## [19] "O2Hb"             "COHb"             "MetHb"
## [22] "O2_suppl"
```

```
glimpse(data)
```

```
## Rows: 163
## Columns: 22
## $ Animal_ID         <chr> "I1", "I1", "I1", "I1", "I1", "I1", "I1", "I1", "...
## $ Trial             <dbl> 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5...
## $ Dpat              <dbl> 0, 0, 0, 0, 0, 5, 5, 5, 5, 5, 0, 0, 0, 0, 0, 0, 0...
## $ Time_min          <dbl> 5, 10, 15, 20, 30, 5, 10, 15, 20, 30, 5, 10, 15, ...
## $ l_star            <dbl> 11.3, 12.9, 13.5, 14.6, 15.6, 12.6, 13.6, 13.8, 1...
## $ a_star            <dbl> 24.5, 31.5, 31.4, 34.1, 34.8, 30.7, 33.0, 31.4, 3...
## $ b_star            <dbl> 11.5, 17.5, 14.7, 19.9, 20.1, 16.5, 18.1, 18.2, 2...
## $ PaO2              <dbl> 27.9, 40.4, 35.3, 41.7, 40.3, 33.9, 34.9, 36.5, 4...
## $ cal_SaO2          <dbl> 45.6, 71.0, 62.3, 73.4, 71.9, 56.7, 60.1, 61.5, 7...
## $ PaCO2             <dbl> 54.9, 55.6, 58.5, 56.3, 57.6, 63.4, 62.4, 63.8, 6...
## $ Cl                <dbl> 113, 113, 111, 111, 109, 107, 106, 106, 105, 105,...
## $ HCO3              <dbl> 27.9, 29.9, 31.3, 31.3, 33.0, 31.0, 32.0, 31.0, 3...
## $ Temp              <dbl> 39.2, 39.7, 39.7, 39.7, 39.6, 39.2, 39.6, 39.6, 3...
## $ pH                <dbl> 7.31, 7.34, 7.34, 7.35, 7.37, 7.30, 7.32, 7.29, 7...
## $ Cartridge_expired <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
## $ Coox_SaO2         <dbl> 53.0, 67.8, 63.3, 74.1, 73.1, 53.8, 64.2, 56.2, 7...
## $ Hct               <dbl> 23, 24, 21, 20, 18, 27, 23, 26, 20, 19, 21, 20, 1...
## $ total_Hb          <dbl> 8.7, 7.8, 7.6, 7.2, 6.7, 9.6, 8.9, 9.8, 7.7, 7.3,...
## $ O2Hb              <dbl> 51.0, 65.4, 60.0, 71.0, 69.4, 51.9, 61.7, 54.2, 6...
## $ COHb              <dbl> 0.0, 1.2, 2.4, 1.0, 1.7, 0.0, 0.8, 0.0, 1.5, 1.4,...
## $ MetHb             <dbl> 3.7, 2.3, 2.8, 3.2, 3.3, 3.4, 3.1, 3.4, 2.4, 2.9,...
```

```
## $ O2_suppl          <chr> "No", "No", "No", "No", "No", "No", "No", "No", "...
```

```r
data[, !(names(data) %in% c('Trial'))] %>%
    select_if(is.numeric) %>%
    my_skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	163
Number of columns	18
Column type frequency:	
numeric	18
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Dpat	0	1.00	10.67	22.33	0.00	0.00	0.00	5.00	70.0
Time_min	0	1.00	22.37	14.21	5.00	10.00	20.00	30.00	50.0
l_star	0	1.00	17.53	2.78	9.20	16.00	18.10	19.40	22.0
a_star	0	1.00	38.97	5.58	19.80	35.90	40.00	42.55	47.8
b_star	0	1.00	24.95	5.75	7.50	21.35	25.50	28.80	36.0
PaO2	7	0.96	85.66	73.43	7.20	44.00	56.55	70.32	311.3
cal_SaO2	15	0.91	82.25	16.17	15.50	74.25	86.80	93.17	99.9
PaCO2	6	0.96	52.57	8.50	33.50	46.20	51.90	57.60	79.6
Cl	6	0.96	110.18	3.14	105.00	108.00	110.00	112.00	121.0
HCO3	14	0.91	29.64	3.27	22.30	27.40	29.30	32.00	37.3
Temp	7	0.96	39.01	0.83	37.00	38.70	39.10	39.50	41.2
pH	14	0.91	7.36	0.07	7.11	7.32	7.37	7.41	7.5
Coox_SaO2	0	1.00	82.49	14.76	16.30	76.20	87.50	92.00	99.3
Hct	8	0.95	23.87	4.52	15.00	20.00	24.00	26.00	38.0
total_Hb	0	1.00	8.86	1.49	5.70	7.70	8.80	9.60	13.4
O2Hb	0	1.00	79.39	14.08	15.70	73.75	83.90	88.50	98.1
COHb	0	1.00	1.79	1.15	0.00	0.90	1.70	2.60	4.7
MetHb	0	1.00	1.98	0.68	0.30	1.45	2.00	2.40	3.8

4 Prepare data

```r
# Convert Animal ID to a factor
data %<>%
    mutate(Animal_ID = factor(Animal_ID,
                    levels = c('I1', 'I2', 'I3', 'I4', 'I5',
                                'I6', 'I8', 'I9', 'I10', 'I11')))
```

5 L*

5.1 Exploratory plots

5.1.1 Linear model

```r
# Composite plot
ggplot(data = data) +
    aes(y = l_star,
        x = Coox_SaO2,
        fill = Animal_ID) +
    geom_point(shape = 21,
               size = 3) +
    geom_smooth(se = FALSE,
                size = 0.8,
                method = 'lm',
                fill = 'blue') +
    scale_fill_viridis_d(name = 'Animal') +
    labs(title = 'SaO2 vs L*',
         subtitle = 'All animals together\nRegression line is simple linear model',
         x = 'SaO2 (%)',
         y = 'L*')
```


SaO2 vs L*

All animals together
Regression line is simple linear model

```r
# Small multiples
ggplot(data = data) +
    aes(y = l_star,
        x = Coox_SaO2) +
    geom_point(shape = 21,
               size = 3,
               fill = '#FFFFFF') +
    geom_smooth(se = FALSE,
                size = 0.8,
                method = 'lm') +
    labs(title = 'SaO2 vs L*',
         subtitle = 'Facetted by animal\nRegression line: linear',
         y = 'L*',
         x = 'SaO2 (%)') +
    facet_wrap(~ Animal_ID,
               ncol = 3)
```

Sa02 vs L*

Facetted by animal
Regression line: linear

5.1.2 Quadratic model
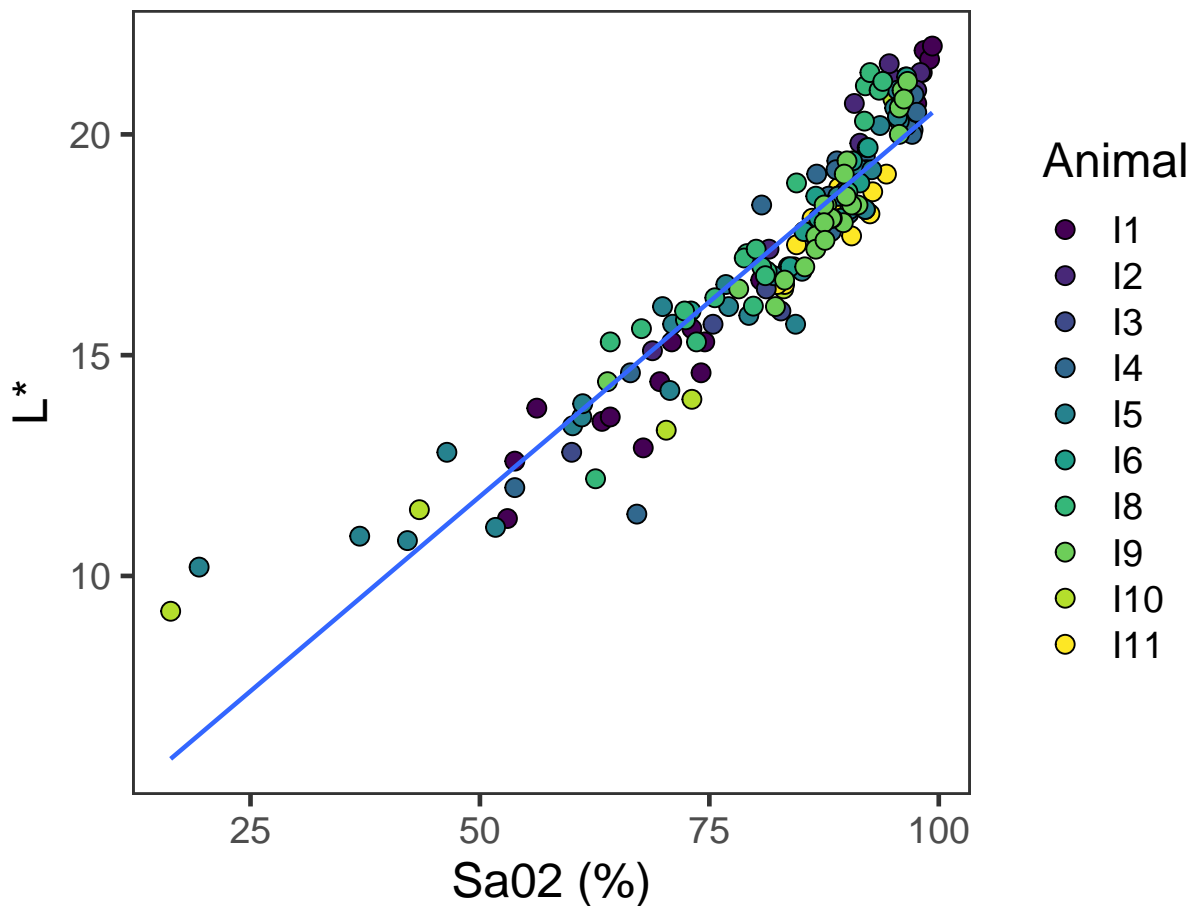
```
# Composite plot
ggplot(data = data) +
    aes(y = l_star,
        x = Coox_SaO2,
        fill = Animal_ID) +
    geom_point(shape = 21,
            size = 3) +
```

```
        geom_smooth(se = FALSE,
                    size = 0.8,
                    method = 'lm',
                    fill = 'blue',
                    formula = y ~ poly(x, 2)) +
        scale_fill_viridis_d(name = 'Animal') +
        labs(title = 'Sa02 vs L*',
             subtitle = 'All animals together\nRegression line is a quadratic polynomial model',
             x = 'Sa02 (%)',
             y = 'L*')
```

Sa02 vs L*

All animals together
Regression line is a quadratic polynomial model
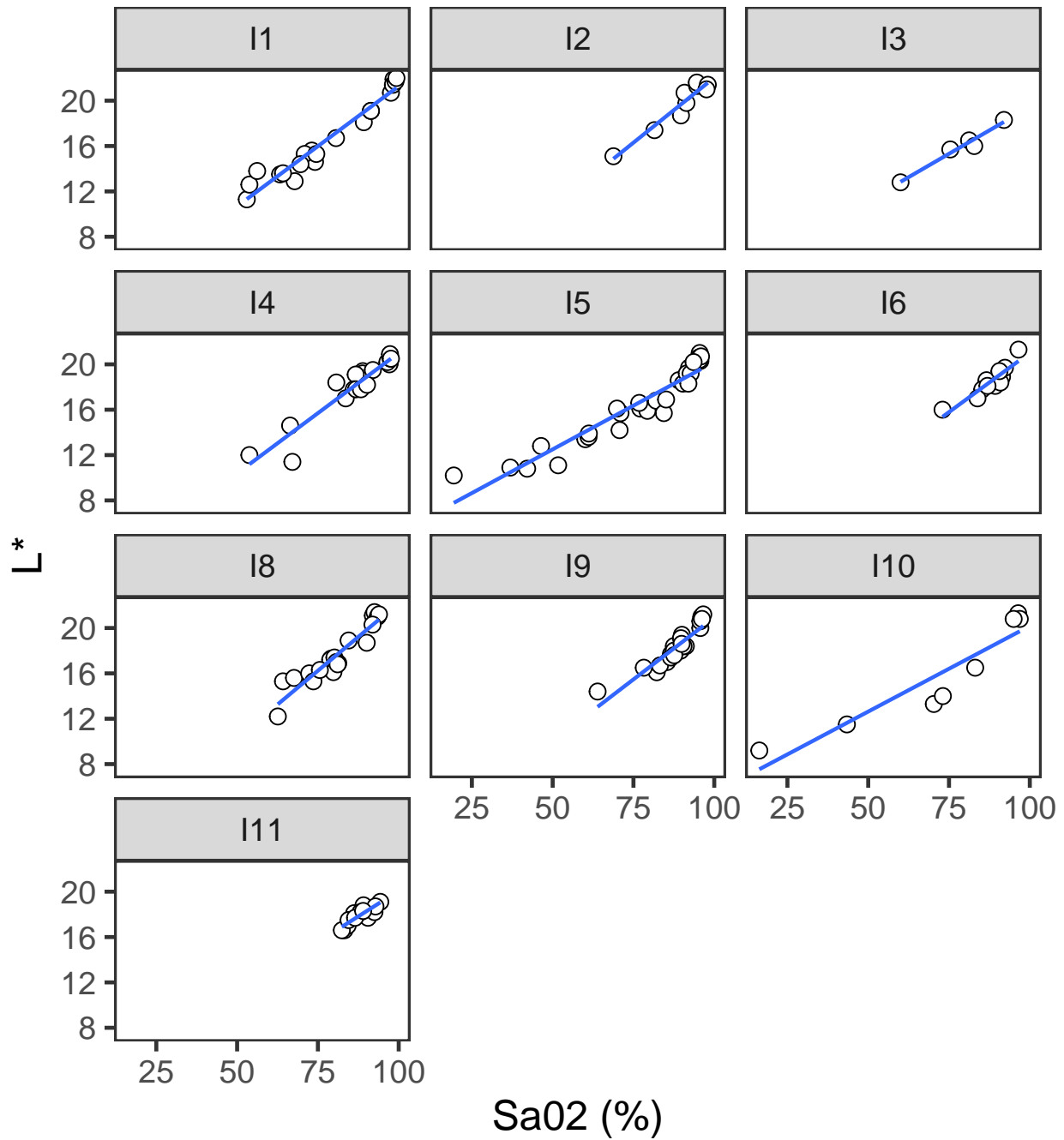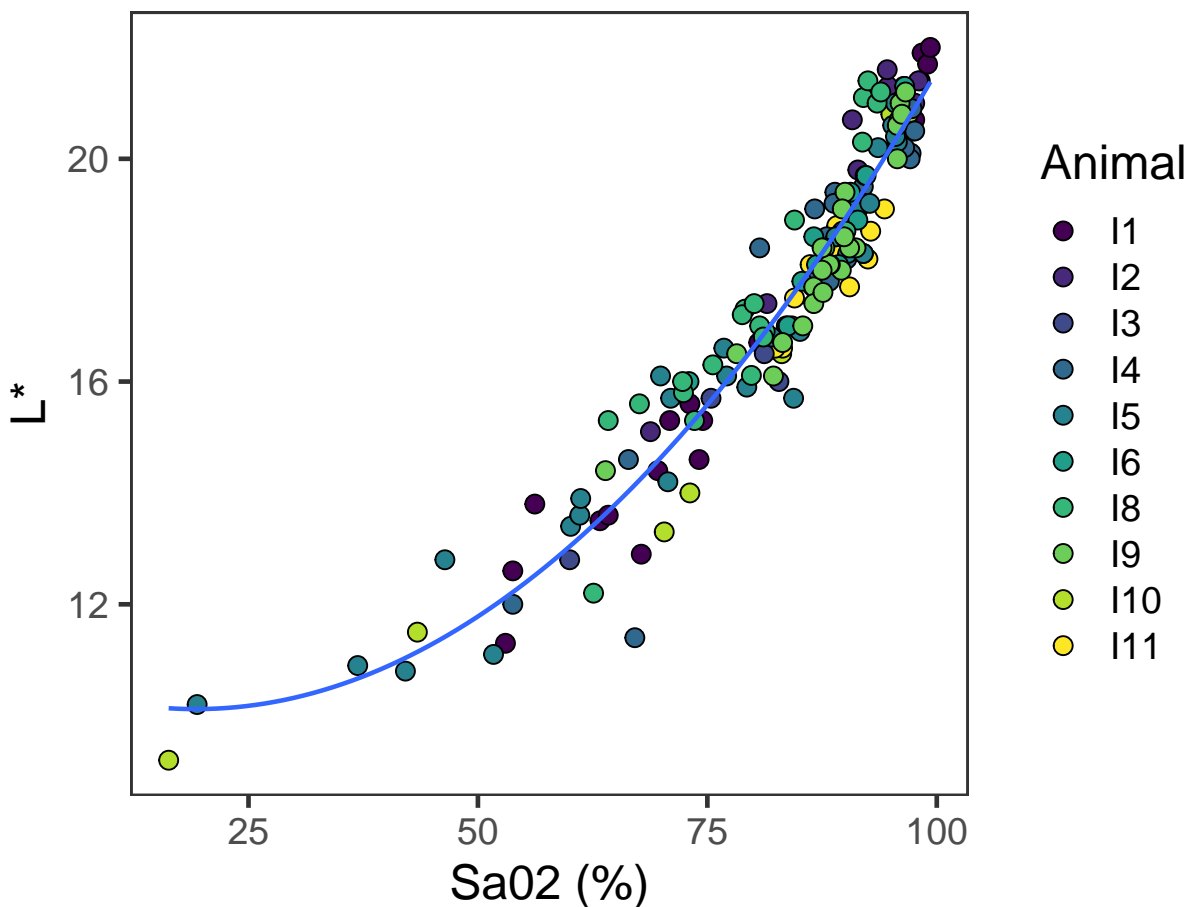


```
# Small multiples
ggplot(data = data) +
    aes(y = l_star,
        x = Coox_Sa02) +
    geom_point(shape = 21,
               size = 3,
               fill = '#FFFFFF') +
    geom_smooth(se = FALSE,
                size = 0.8,
                colour = 'blue',
                method = 'lm',
                formula = y ~ poly(x, 2, raw = TRUE)) +
    labs(title = 'Sa02 vs L*',
```
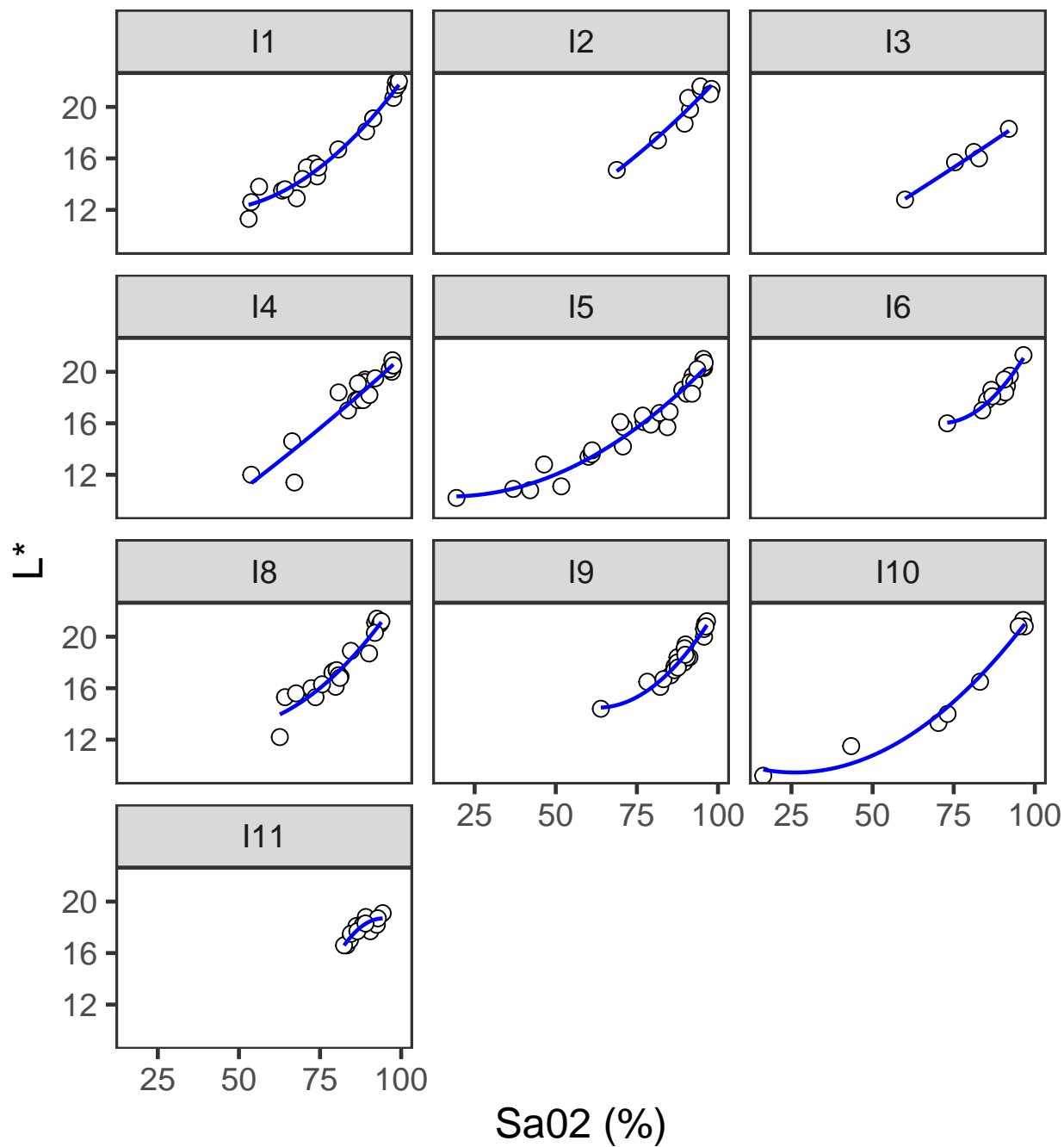
```
        subtitle = 'Facetted by animal\nRegression line: quadratic polynomial',
        y = 'L*',
        x = 'Sa02 (%)') +
    facet_wrap(~ Animal_ID,
            ncol = 3)
```

Sa02 vs L*

Facetted by animal
Regression line: quadratic polynomial

5.2   Model selection

Model selection based on the outcome of likelihood ratio test of goodness of fit.

```
# Null model
lstar_null <- lmer(l_star ~ 1 + (1 | Animal_ID),
                   data = data,
                   REML = FALSE)

## First-order polynomial (linear) model
lstar_linear <- lmer(l_star ~ Coox_SaO2 + (1 | Animal_ID),
                     data = data,
                     REML = FALSE)

## Second-order polynomial (quadratic) model
lstar_poly <- lmer(l_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID),
                   data = data,
                   REML = FALSE)

# Compare model fits
anova(lstar_null, lstar_linear, lstar_poly, test = 'LRT')
```

```
## Data: data
## Models:
## lstar_null: l_star ~ 1 + (1 | Animal_ID)
## lstar_linear: l_star ~ Coox_SaO2 + (1 | Animal_ID)
## lstar_poly: l_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID)
##              Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lstar_null    3 798.66 807.94 -396.33   792.66
## lstar_linear  4 463.12 475.50 -227.56   455.12 337.53      1  < 2.2e-16 ***
## lstar_poly    5 361.75 377.22 -175.88   351.75 103.37      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Best fit model is the second-order polynomial (quadratic) model.

5.3   Explore the best fit model

```
# Summary output
summary(lstar_poly)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: l_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##    361.8    377.2   -175.9    351.8      158
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7615 -0.7203  0.0194  0.6264  2.6754
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Animal_ID (Intercept) 0.1253   0.3540
##  Residual              0.4589   0.6774
## Number of obs: 163, groups:  Animal_ID, 10
##
## Fixed effects:
##                    Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)         17.5035     0.1266  9.2565  138.29   <2e-16 ***
```

```
## poly(Coox_SaO2, 2)1  33.3450      0.7193 160.2496    46.36    <2e-16 ***
## poly(Coox_SaO2, 2)2   8.5646      0.7048 159.1582    12.15    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) p(C_SO2,2)1
## p(C_SO2,2)1 -0.005
## p(C_SO2,2)2 -0.008 -0.015
```

```r
# Check model assumptions
plot_model(model = lstar_poly,
           type = 'diag')[2:4]
```
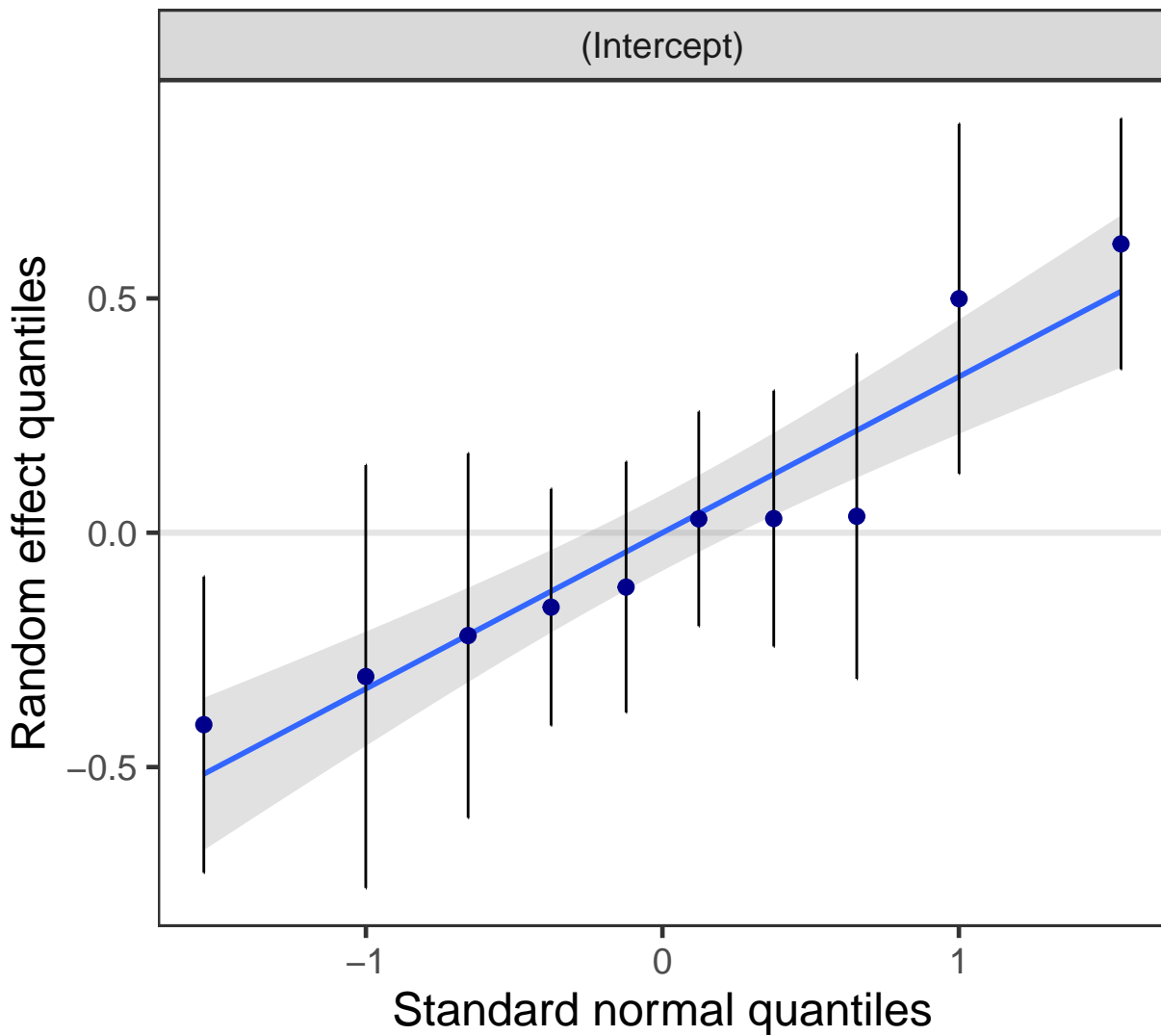
```
## [[1]]
## [[1]]$Animal_ID
```



```
##
##
## [[2]]
```

Non−normality of residuals
Distribution should look like normal curve


```
##
## [[3]]
```
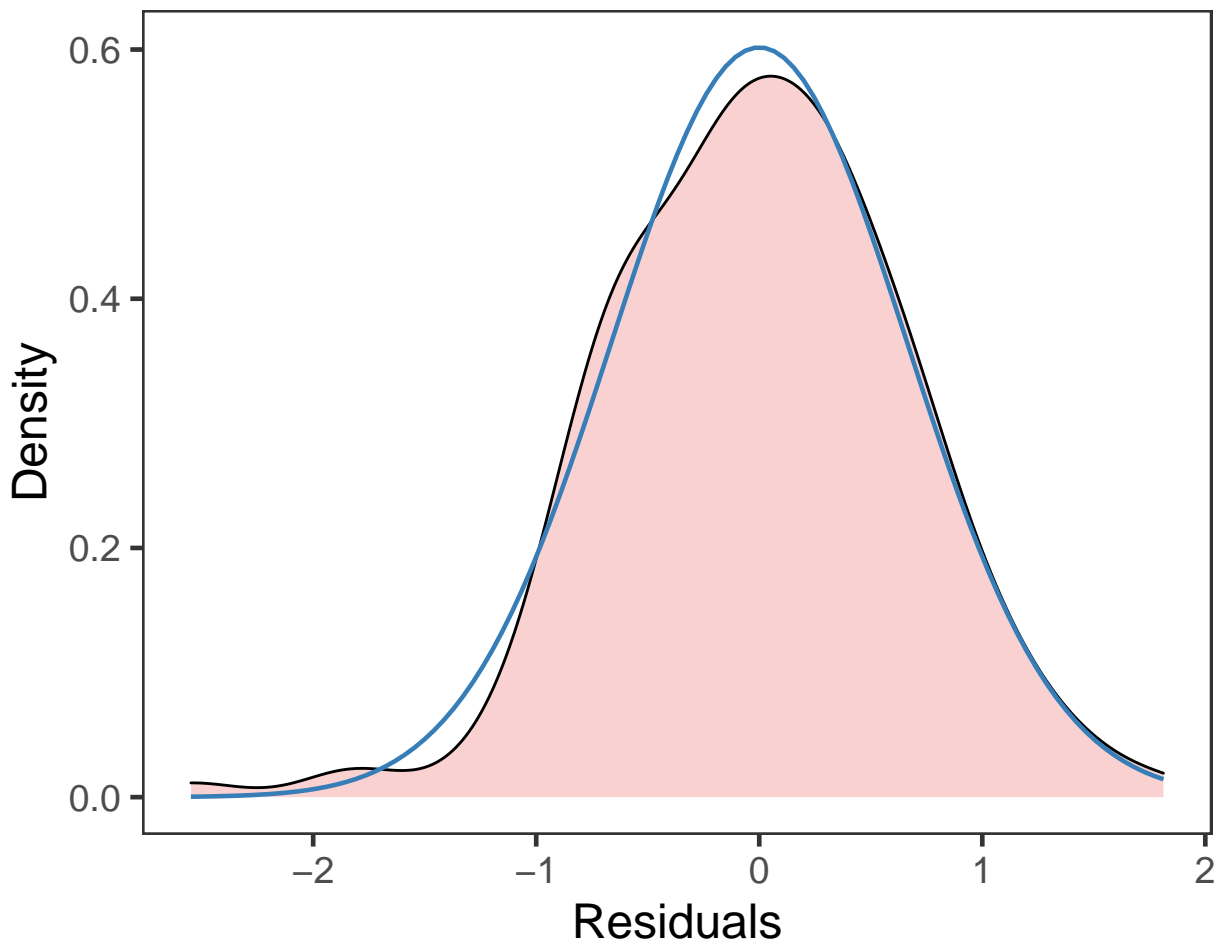
Homoscedasticity (constant variance of residuals)

Amount and distance of points scattered above/below line is equal
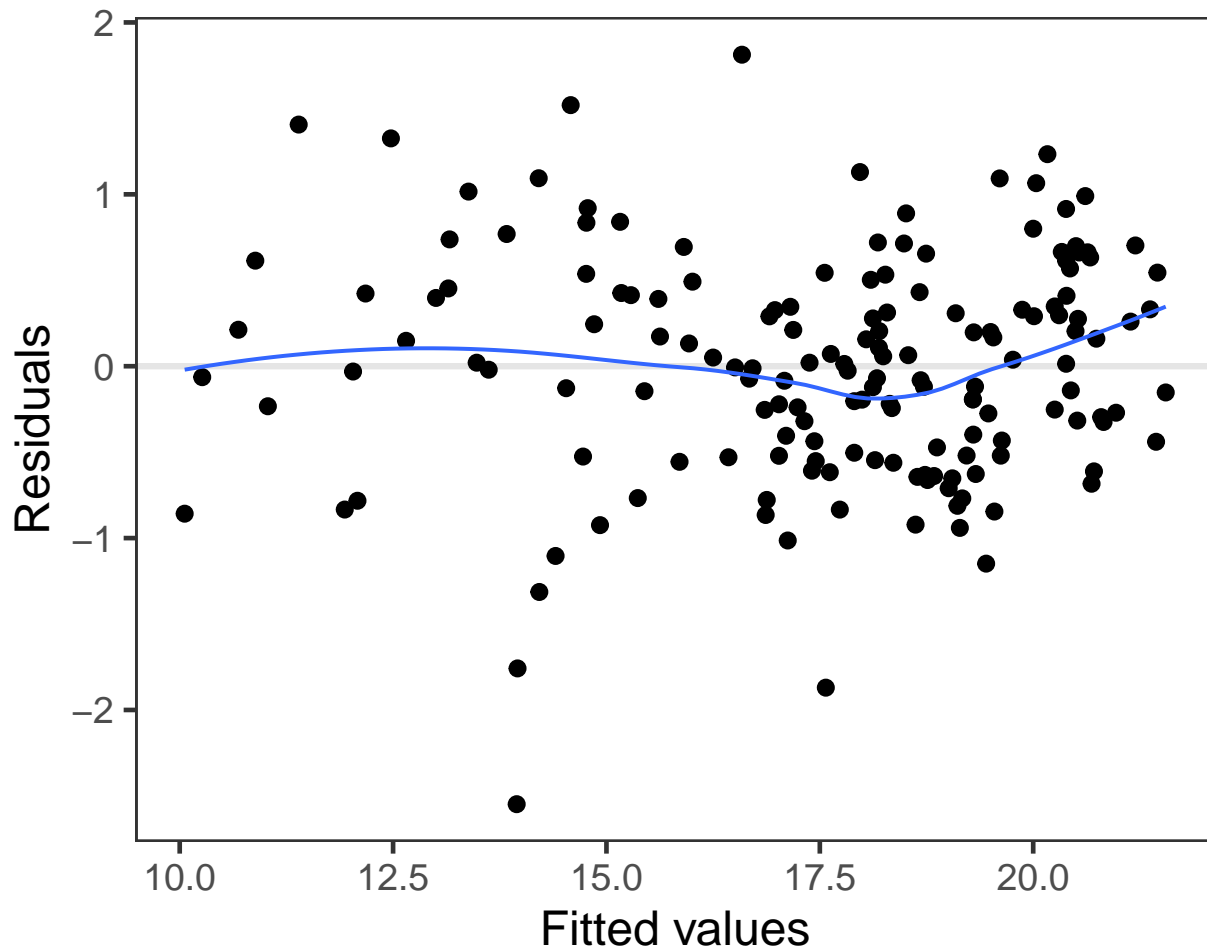

```
# R-squared (based on likelihood ratio test)
r.squaredLR(lstar_poly, null = lstar_null)
```

```
## [1] 0.9331251
## attr(,"adj.r.squared")
## [1] 0.9403924
```

```
# Root mean square error
RMSE.merMod(lstar_poly)
```

```
## [1] 0.6609619
```

5.4 Generate prediction intervals

```
# Prediction interval
lstar_lqmm <- lqmm(fixed = l_star ~ poly(Coox_SaO2, 2),
            random = ~ 1,
            group = ~ Animal_ID,
            tau = c(0.025, 0.976),
            data = data)

lstar_interval <- as.data.frame(predict(lstar_lqmm, level = 0))

names(lstar_interval) <- paste0('Q', c(2.5, 97.5))
```

```r
lstar_interval <- cbind(data, lstar_interval)

# Numeric output
lstar_interval %>%
    dplyr::select(Coox_SaO2, l_star, Q2.5, Q97.5) %>%
    arrange(Coox_SaO2) %>%
    unique(.) %>%
    kable(caption = 'L*: 95% prediction interval',
          col.names = c('SaO2 (%)', 'L*', 'Q2.5', 'Q97.5'))
```

Table 3: L*: 95% prediction interval

	SaO2 (%)	L*	Q2.5	Q97.5
1	16.3	9.20	8.841643	11.22843
2	19.4	10.20	8.816442	11.23688
3	36.9	10.90	9.320718	11.89098
4	42.1	10.80	9.682259	12.28390
5	43.4	11.50	9.787799	12.39635
6	46.4	12.80	10.054488	12.67754
7	51.7	11.10	10.604532	13.24830
8	53.0	11.30	10.754836	13.40273
9	53.8	12.60	10.850344	13.50059
10	53.8	12.00	10.850344	13.50059
11	56.2	13.80	11.150641	13.80709
12	60.0	12.80	11.668366	14.33201
13	60.1	13.40	11.682690	14.34648
14	61.1	13.60	11.827900	14.49303
15	61.2	13.90	11.842619	14.50787
16	62.6	12.20	12.052441	14.71916
17	63.3	13.50	12.159989	14.82728
18	63.9	14.40	12.253572	14.92126
19	64.2	13.60	12.300848	14.96871
20	64.2	15.30	12.300848	14.96871
21	66.4	14.60	12.657399	15.32590
22	67.1	11.40	12.774488	15.44297
23	67.6	15.60	12.859199	15.52759
24	67.8	12.90	12.893335	15.56168
25	68.8	15.10	13.066164	15.73413
26	69.6	14.40	13.207011	15.87452
27	69.9	16.10	13.260420	15.92772
28	70.3	13.30	13.332134	15.99912
29	70.7	14.20	13.404423	16.07106
30	70.9	15.30	13.440782	16.10724
31	71.0	15.70	13.459015	16.12537
32	72.3	16.00	13.699315	16.36421
33	72.4	15.80	13.718051	16.38282
34	73.0	16.00	13.831218	16.49518
35	73.1	15.60	13.850204	16.51402
36	73.1	14.00	13.850204	16.51402
37	73.6	15.30	13.945676	16.60874
38	74.1	14.60	14.042045	16.70430
39	74.5	15.30	14.119785	16.78136
40	75.4	15.70	14.296799	16.95670
41	75.6	16.30	14.336530	16.99604
42	76.8	16.60	14.577930	17.23487
43	77.1	16.10	14.639086	17.29534
44	78.2	16.50	14.866090	17.51964
45	78.8	17.20	14.991740	17.64370

	Sa02 (%)	L*	Q2.5	Q97.5
46	79.1	17.30	15.055049	17.70619
47	79.3	15.90	15.097434	17.74801
48	79.8	16.11	15.204025	17.85316
49	80.1	17.40	15.268410	17.91666
50	80.6	16.70	15.376436	18.02315
51	80.7	18.40	15.398149	18.04455
52	80.7	17.00	15.398149	18.04455
53	81.1	16.80	15.485359	18.13049
54	81.2	16.50	15.507251	18.15206
55	81.3	16.90	15.529179	18.17366
56	81.5	17.40	15.573143	18.21696
57	82.0	16.80	15.683679	18.32582
58	82.2	16.10	15.728145	18.36959
59	82.4	16.60	15.772754	18.41350
60	82.8	16.00	15.862403	18.50173
61	83.1	16.50	15.930016	18.56825
62	83.2	16.60	15.952626	18.59049
63	83.2	16.70	15.952626	18.59049
64	83.6	17.00	16.043423	18.67979
65	83.8	17.00	16.089036	18.72464
66	84.2	17.00	16.180694	18.81475
67	84.4	15.70	16.226738	18.86001
68	84.5	17.50	16.249813	18.88269
69	84.5	18.90	16.249813	18.88269
70	85.1	16.90	16.389022	19.01947
71	85.3	17.80	16.435711	19.06533
72	85.4	17.00	16.459110	19.08831
73	86.1	17.80	16.623904	19.25012
74	86.2	18.10	16.647590	19.27337
75	86.5	17.70	16.718862	19.34333
76	86.6	18.60	16.742691	19.36671
77	86.6	17.70	16.742691	19.36671
78	86.6	17.40	16.742691	19.36671
79	86.7	19.10	16.766556	19.39013
80	86.8	17.80	16.790457	19.41358
81	86.9	18.10	16.814393	19.43707
82	87.5	18.40	16.958766	19.57869
83	87.5	18.00	16.958766	19.57869
84	87.6	17.60	16.982954	19.60241
85	87.8	18.40	17.031437	19.64996
86	88.0	18.60	17.080064	19.69764
87	88.2	18.20	17.128834	19.74545
88	88.3	17.80	17.153273	19.76941
89	88.3	18.10	17.153273	19.76941
90	88.4	18.10	17.177747	19.79340
91	88.8	18.30	17.276005	19.88970
92	88.8	19.20	17.276005	19.88970
93	88.9	19.40	17.300659	19.91385
94	89.0	18.30	17.325349	19.93805
95	89.0	18.60	17.325349	19.93805
97	89.1	18.80	17.350075	19.96227
98	89.2	18.10	17.374836	19.98653
99	89.3	18.10	17.399634	20.01083
100	89.6	18.00	17.474242	20.08391
101	89.7	18.70	17.499182	20.10834
102	89.7	19.10	17.499182	20.10834
103	89.9	18.60	17.549172	20.15729

	Sa02 (%)	L*	Q2.5	Q97.5
104	90.0	19.40	17.574221	20.18182
105	90.1	18.70	17.599305	20.20639
106	90.2	18.20	17.624425	20.23098
107	90.3	18.30	17.649582	20.25561
108	90.5	17.70	17.700002	20.30497
109	90.5	18.40	17.700002	20.30497
110	90.6	19.40	17.725265	20.32970
111	90.8	20.70	17.775901	20.37927
112	90.9	18.40	17.801272	20.40410
113	91.2	18.40	17.877602	20.47879
114	91.4	19.10	17.928667	20.52876
116	91.4	19.80	17.928667	20.52876
117	91.4	18.90	17.928667	20.52876
118	91.5	19.20	17.954254	20.55379
119	91.9	20.30	18.056959	20.65427
120	92.0	18.30	18.082725	20.67947
121	92.0	19.50	18.082725	20.67947
123	92.0	21.10	18.082725	20.67947
124	92.1	19.20	18.108527	20.70470
125	92.2	19.70	18.134365	20.72997
126	92.3	19.70	18.160238	20.75527
127	92.5	18.20	18.212093	20.80598
128	92.5	21.40	18.212093	20.80598
129	92.7	19.20	18.264091	20.85682
130	92.8	18.70	18.290144	20.88229
131	93.5	21.00	18.473519	21.06153
132	93.6	20.20	18.499859	21.08727
133	93.9	21.20	18.579093	21.16469
134	94.3	19.10	18.685242	21.26839
135	94.6	21.60	18.765230	21.34652
136	94.7	21.30	18.791965	21.37263
137	95.0	20.80	18.872383	21.45117
138	95.2	20.60	18.926175	21.50369
139	95.5	21.00	19.007131	21.58273
140	95.5	20.40	19.007131	21.58273
141	95.7	20.30	19.061282	21.63559
142	95.7	20.00	19.061282	21.63559
143	95.7	20.60	19.061282	21.63559
144	95.9	20.70	19.115575	21.68858
145	96.0	21.00	19.142776	21.71513
146	96.2	20.80	19.197285	21.76832
147	96.4	21.30	19.251938	21.82165
148	96.5	20.20	19.279318	21.84837
149	96.5	21.30	19.279318	21.84837
150	96.6	21.20	19.306734	21.87512
151	96.9	20.80	19.389197	21.95557
152	97.1	20.00	19.444352	22.00937
153	97.2	20.10	19.471983	22.03632
154	97.3	20.90	19.499650	22.06331
155	97.5	20.50	19.555091	22.11738
156	97.6	20.70	19.582866	22.14446
157	97.6	21.00	19.582866	22.14446
158	97.6	20.50	19.582866	22.14446
159	98.0	21.40	19.694323	22.25314
160	98.2	21.40	19.750267	22.30768
161	98.4	21.90	19.806354	22.36236
162	99.0	21.70	19.975477	22.52720

	Sa02 (%)	L*	Q2.5	Q97.5
163	99.3	22.00	20.060523	22.61007

```
# Plot output
ggplot(data = lstar_interval) +
    aes(x = Coox_SaO2) +
    geom_point(data = data,
            aes(y = l_star,
                x = Coox_SaO2,
                fill = Animal_ID),
            shape = 21,
            size = 3) +
    geom_smooth(data = lstar_interval,
            aes(y = `Q2.5`),
            se = FALSE) +
    geom_smooth(data = lstar_interval,
            aes(y = `Q97.5`),
            se = FALSE) +
    scale_fill_viridis_d(name = 'Animal') +
    labs(title = 'Sa02 vs L*',
        subtitle = '95% prediction interval',
        y = 'L*',
        x = 'Sa02 (%)')
```


Sa02 vs L*

95% prediction interval
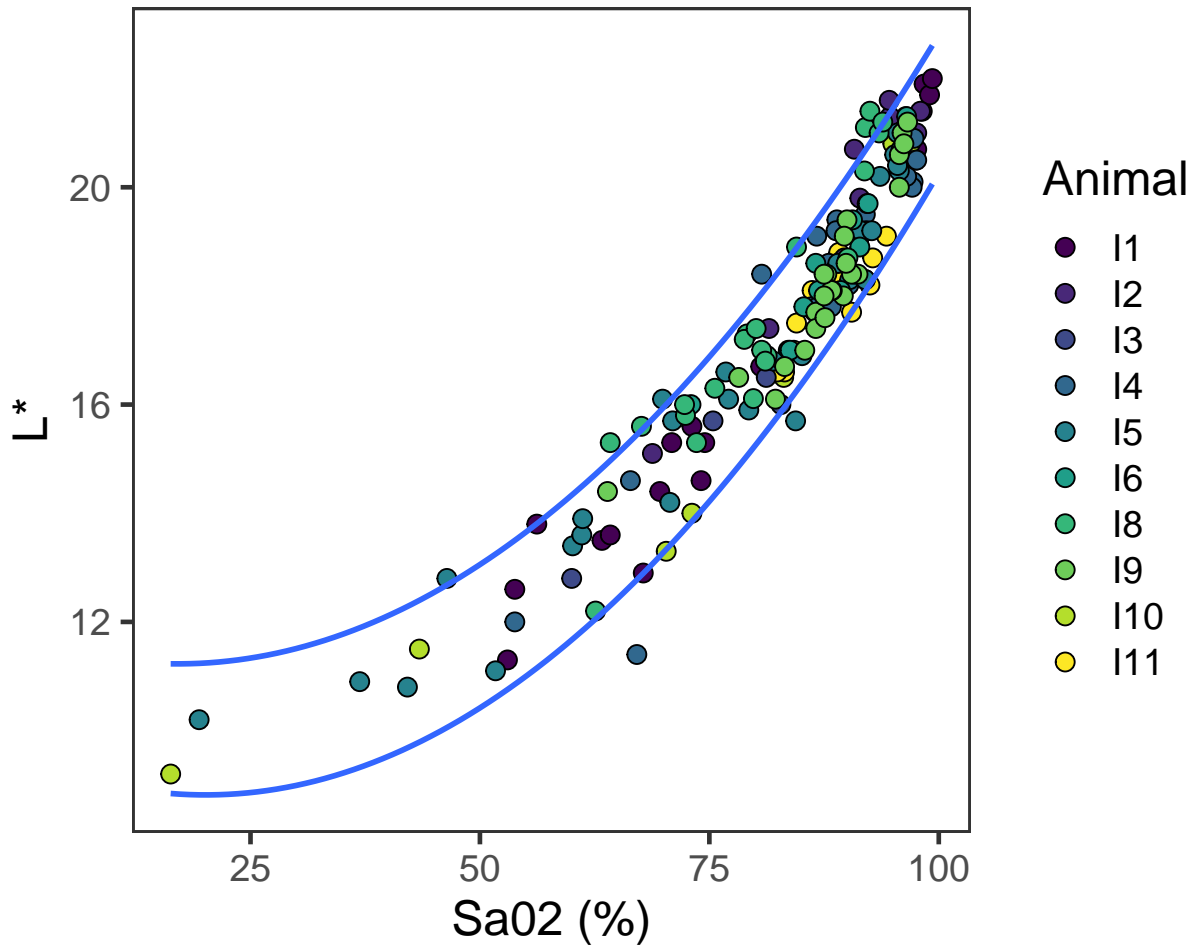
5.5 Generate predicted values

```r
# New data
lstar_grid <- expand.grid(Coox_SaO2 = c(15, 25, 35, 45, 55, 65, 75, 85, 95),
                          Animal_ID = 'I12')

# Generate predicted vs independent variable SaO2
lstar_predicted <- predict(lstar_poly,
                           lstar_grid,
                           allow.new.levels = TRUE)

# Tabulate
data.frame(SaO2 = lstar_grid$Coox_SaO2,
           lstar = lstar_predicted) %>%
    kable(caption = 'L* predicted values',
          col.names = c('SaO2 (%)', 'L*'))
```

Table 4: L* predicted values

SaO2 (%)	L*
15	10.30631
25	10.24610
35	10.55715
45	11.23943
55	12.29297
65	13.71774
75	15.51377
85	17.68103
95	20.21955

6   a*

6.1   Exploratory plots
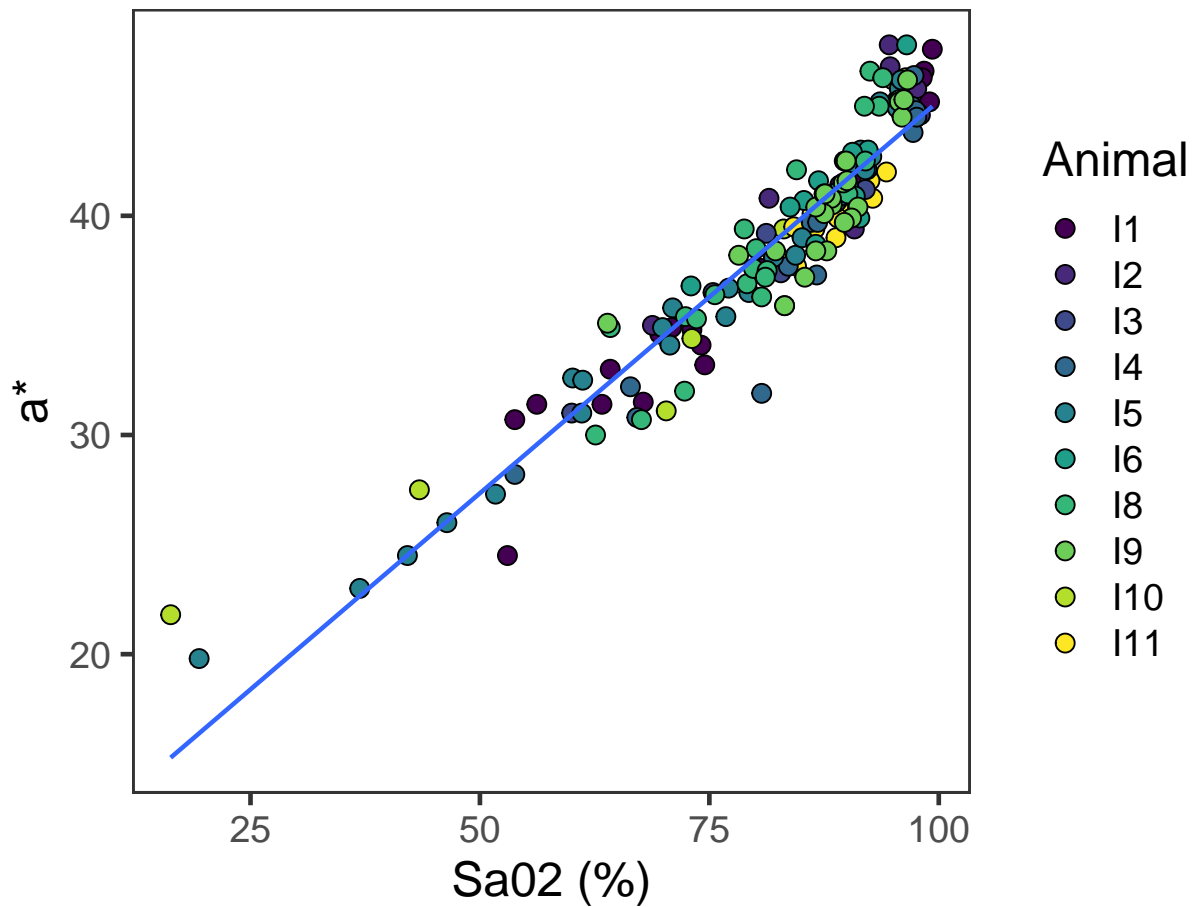
6.1.1   Linear model

```r
# Composite plot
ggplot(data = data) +
    aes(y = a_star,
        x = Coox_SaO2,
        fill = Animal_ID) +
    geom_point(shape = 21,
               size = 3) +
    geom_smooth(se = FALSE,
                size = 0.8,
                method = 'lm',
                fill = 'blue') +
    scale_fill_viridis_d(name = 'Animal') +
    labs(title = 'SaO2 vs a*',
         subtitle = 'All animals together\nRegression line is simple linear model',
         x = 'SaO2 (%)',
         y = 'a*')
```

Sa02 vs a*

All animals together
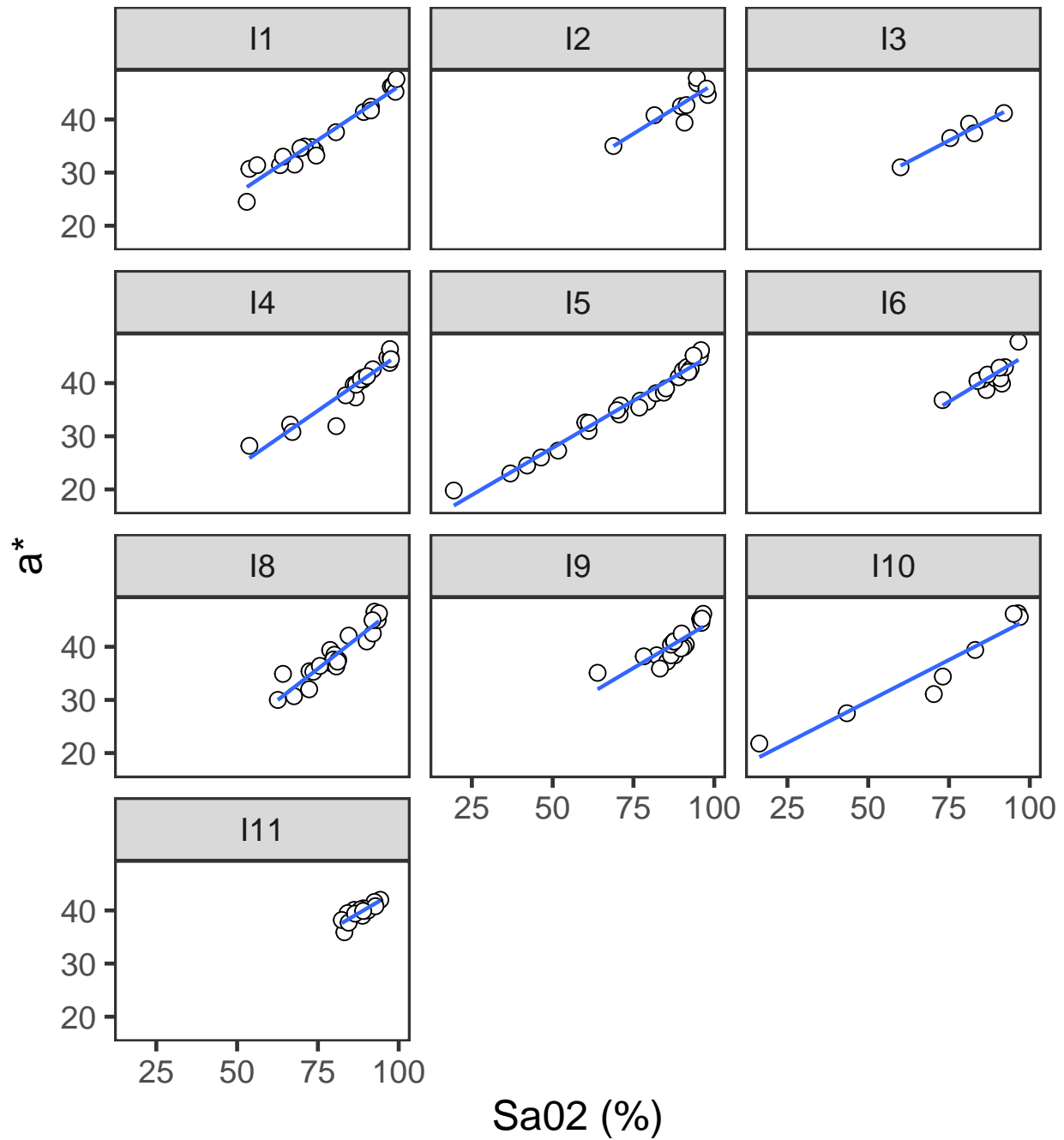Regression line is simple linear model


```r
# Small multiples
ggplot(data = data) +
    aes(y = a_star,
        x = Coox_Sa02) +
    geom_point(shape = 21,
               size = 3,
               fill = '#FFFFFF') +
    geom_smooth(se = FALSE,
                size = 0.8,
                method = 'lm') +
    labs(title = 'Sa02 vs a*',
         subtitle = 'Facetted by animal\nRegression line: linear',
         y = 'a*',
         x = 'Sa02 (%)') +
    facet_wrap(~ Animal_ID,
               ncol = 3)
```

Sa02 vs a*

Facetted by animal
Regression line: linear

6.1.2 Quadratic model
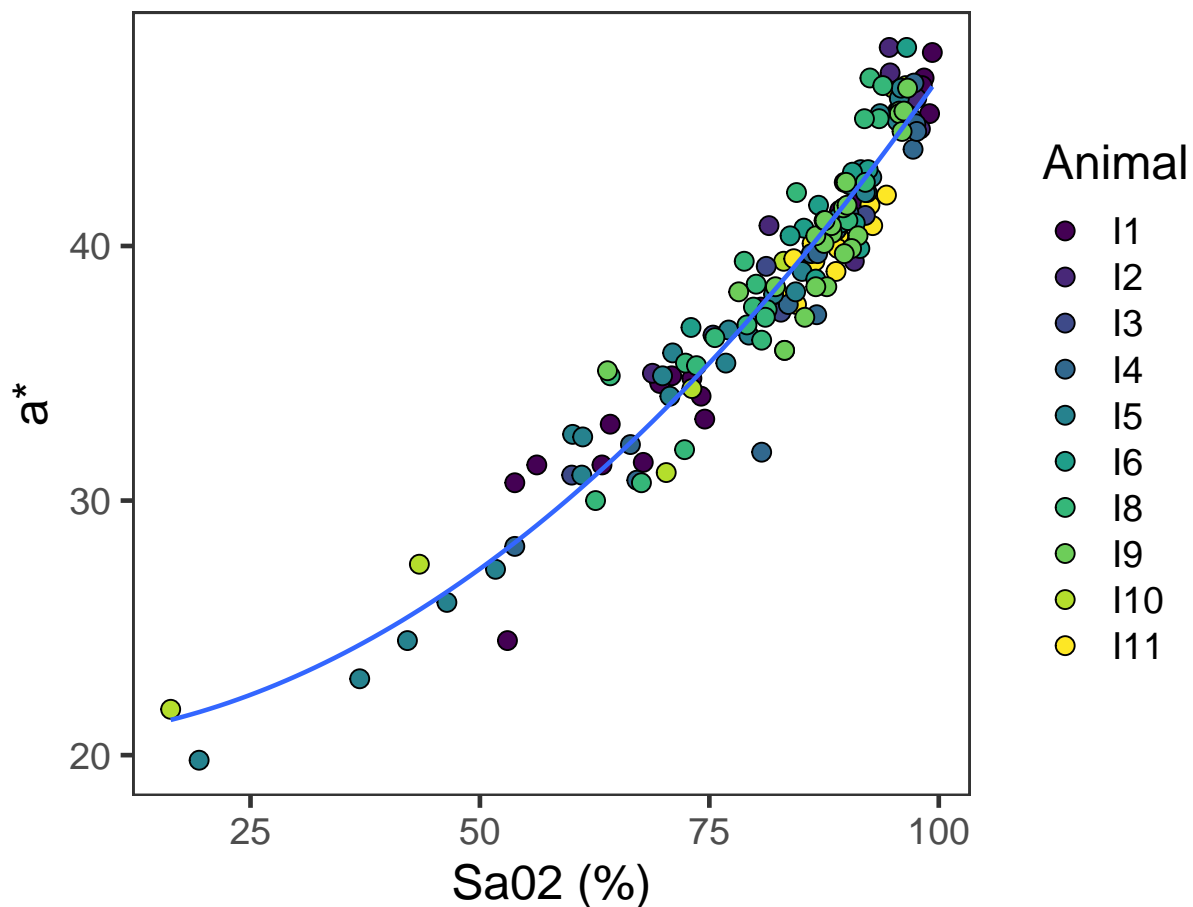
```r
# Composite plot
ggplot(data = data) +
    aes(y = a_star,
        x = Coox_SaO2,
        fill = Animal_ID) +
    geom_point(shape = 21,
               size = 3) +
```

```
            geom_smooth(se = FALSE,
                        size = 0.8,
                        method = 'lm',
                        fill = 'blue',
                        formula = y ~ poly(x, 2)) +
            scale_fill_viridis_d(name = 'Animal') +
            labs(title = 'Sa02 vs a*',
                 subtitle = 'All animals together\nRegression line is a quadratic polynomial model',
                 x = 'Sa02 (%)',
                 y = 'a*')
```

Sa02 vs a*

All animals together
Regression line is a quadratic polynomial model


```
# Small multiples
ggplot(data = data) +
    aes(y = a_star,
        x = Coox_Sa02) +
    geom_point(shape = 21,
               size = 3,
               fill = '#FFFFFF') +
    geom_smooth(se = FALSE,
                size = 0.8,
                colour = 'blue',
                method = 'lm',
                formula = y ~ poly(x, 2, raw = TRUE)) +
    labs(title = 'Sa02 vs a*',
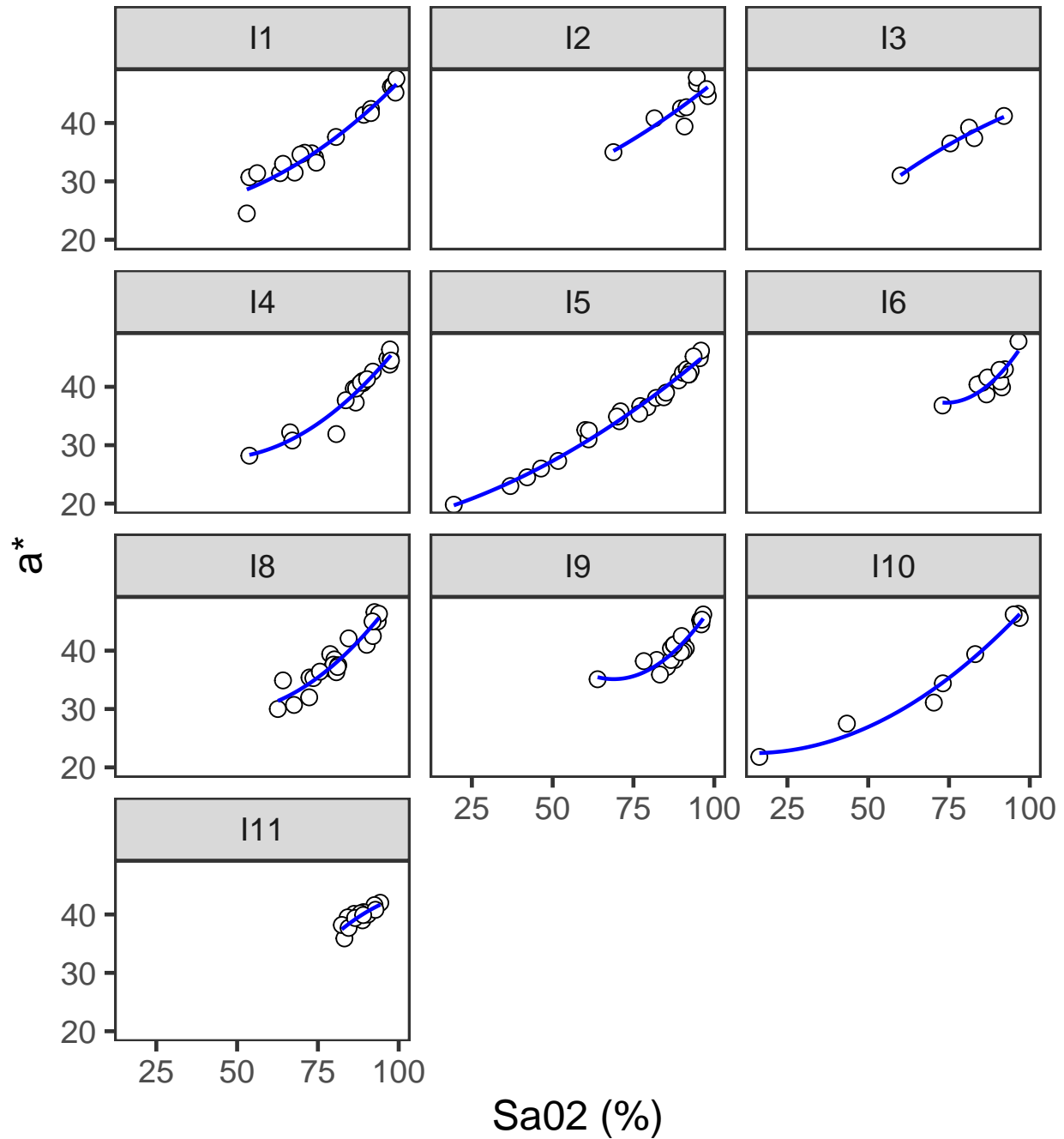```

```
        subtitle = 'Facetted by animal\nRegression line: quadratic polynomial',
        y = 'a*',
        x = 'Sa02 (%)') +
  facet_wrap(~ Animal_ID,
              ncol = 3)
```

Sa02 vs a*

Facetted by animal
Regression line: quadratic polynomial

6.2 Model selection

Model selection based on the outcome of likelihood ratio test of goodness of fit.

```
# Null model
astar_null <- lmer(a_star ~ 1 + (1 | Animal_ID),
                   data = data,
                   REML = FALSE)

## First-order polynomial (linear) model
astar_linear <- lmer(a_star ~ Coox_SaO2 + (1 | Animal_ID),
                   data = data,
                   REML = FALSE)

## Second-order polynomial (quadratic) model
astar_poly <- lmer(a_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID),
                   data = data,
                   REML = FALSE)

# Compare model fits
anova(astar_null, astar_linear, astar_poly, test = 'LRT')
```

```
## Data: data
## Models:
## astar_null: a_star ~ 1 + (1 | Animal_ID)
## astar_linear: a_star ~ Coox_SaO2 + (1 | Animal_ID)
## astar_poly: a_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID)
##               Df     AIC     BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## astar_null    3 1026.63 1035.91 -510.32  1020.63
## astar_linear  4  654.50  666.88 -323.25   646.50 374.13      1  < 2.2e-16 ***
## astar_poly    5  604.82  620.29 -297.41   594.82  51.68      1  6.532e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Best fit model is the second-order polynomial (quadratic) model.

## 6.3 Explore the best fit model

```
# Summary output
summary(astar_poly)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: a_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##    604.8    620.3   -297.4    594.8      158
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5445 -0.6146  0.0390  0.6159  2.7739
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Animal_ID (Intercept) 0.2961   0.5441
##  Residual              2.0994   1.4489
## Number of obs: 163, groups:  Animal_ID, 10
##
## Fixed effects:
##                     Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)           39.007      0.212   9.353 184.003  < 2e-16 ***
```

22

```
## poly(Coox_SaO2, 2)1    68.374        1.524 162.491   44.856  < 2e-16 ***
## poly(Coox_SaO2, 2)2    11.675        1.497 161.244    7.801 7.24e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) p(C_SO2,2)1
## p(C_SO2,2)1 -0.007
## p(C_SO2,2)2 -0.008 -0.012
```

```
# Check model assumptions
plot_model(model = astar_poly,
           type = 'diag')[2:4]
```
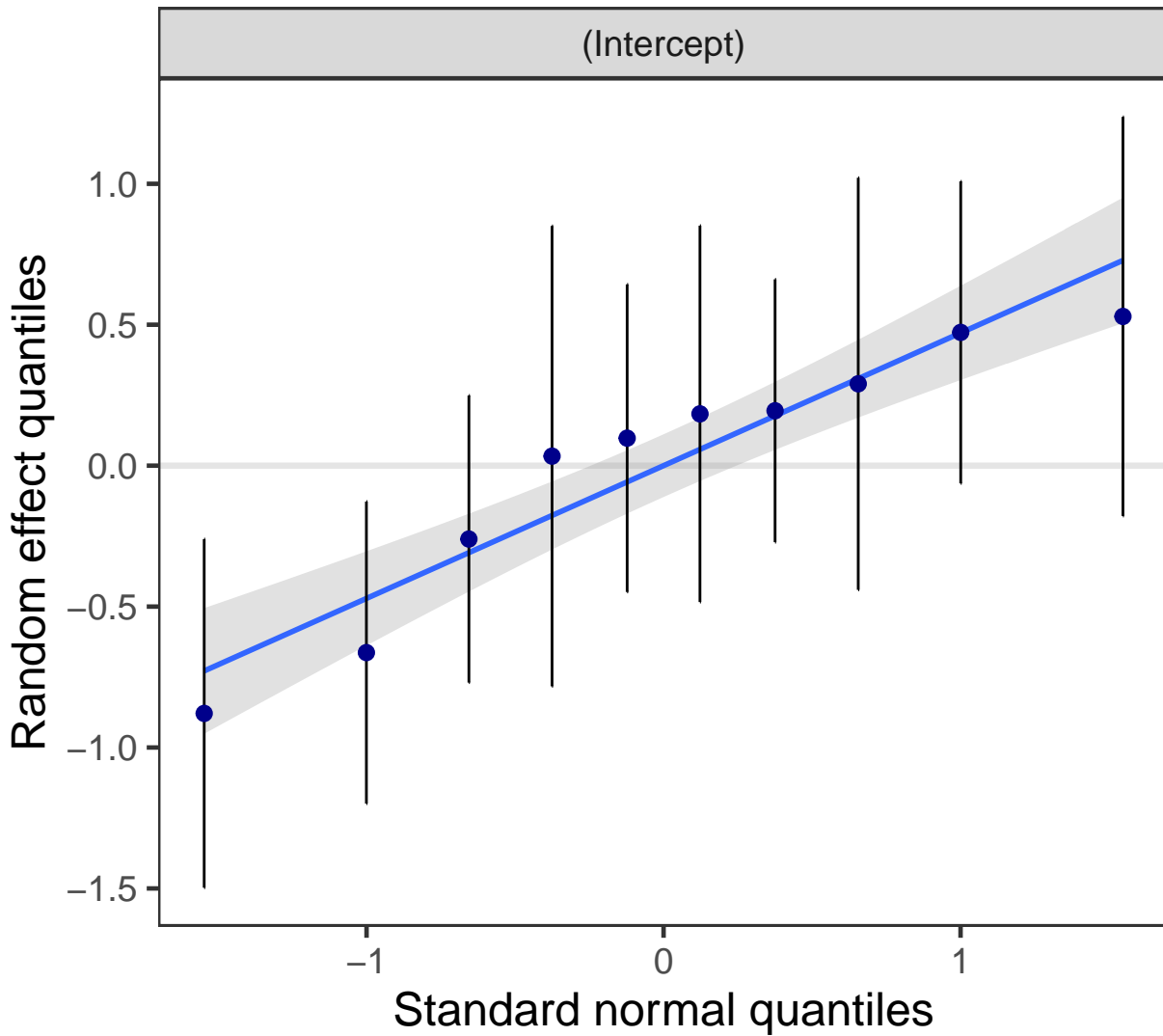
```
## [[1]]
## [[1]]$Animal_ID
```



```
##
##
## [[2]]
```

Non−normality of residuals

Distribution should look like normal curve


```
##
## [[3]]
```
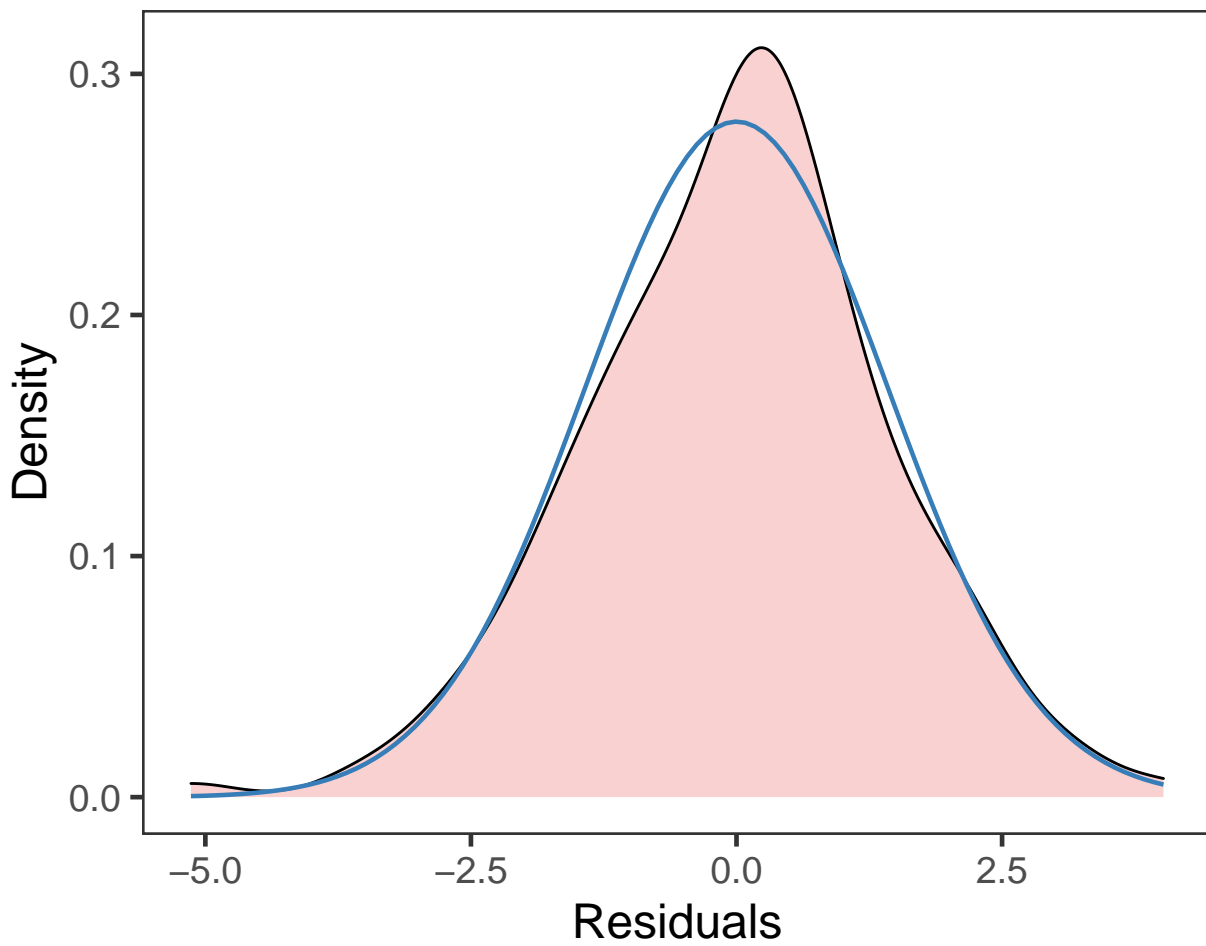
Homoscedasticity (constant variance of residuals)
Amount and distance of points scattered above/below line is equa
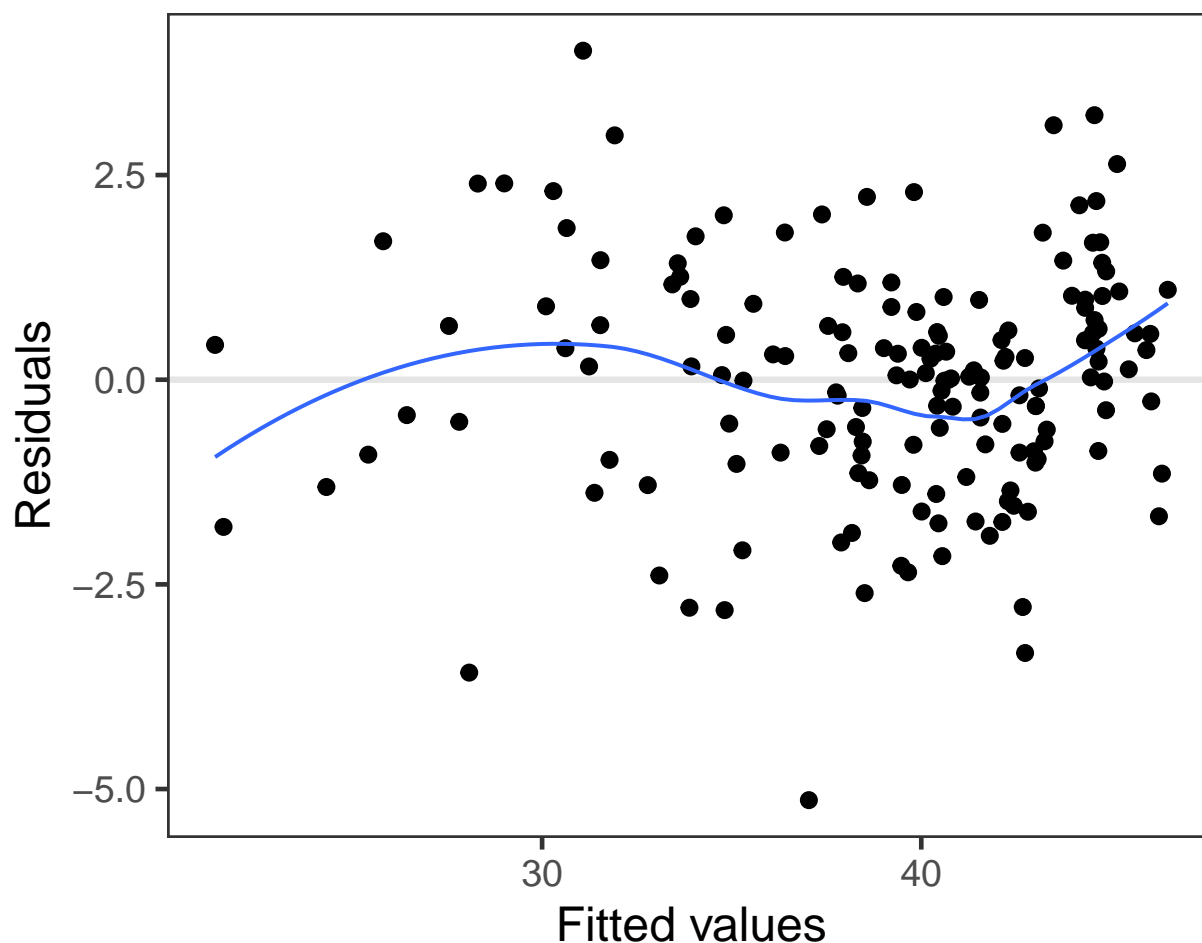
```
# R-squared (based on likelihood ratio test)
r.squaredLR(astar_poly, null = astar_null)
```

```
## [1] 0.9266361
## attr(,"adj.r.squared")
## [1] 0.9284078
```

```
# Root mean square error
RMSE.merMod(astar_poly)
```

```
## [1] 1.419339
```

6.4   Generate prediction intervals

```
# Prediction interval
astar_lqmm <- lqmm(fixed = a_star ~ poly(Coox_SaO2, 2),
                   random = ~ 1,
                   group = ~ Animal_ID,
                   tau = c(0.025, 0.976),
                   data = data)

astar_interval <- as.data.frame(predict(astar_lqmm, level = 0))

names(astar_interval) <- paste0('Q', c(2.5, 97.5))
```

```
astar_interval <- cbind(data, astar_interval)

# Numeric output
astar_interval %>%
    dplyr::select(Coox_SaO2, a_star, Q2.5, Q97.5) %>%
    arrange(Coox_SaO2) %>%
    unique(.) %>%
    kable(caption = 'a*: 95% prediction interval',
          col.names = c('SaO2 (%)', 'a*', 'Q2.5', 'Q97.5'))
```

Table 5: a*: 95% prediction interval

|	SaO2 (%)	a*	Q2.5	Q97.5
1	16.3	21.8	18.48844	24.70193
2	19.4	19.8	18.79642	25.01932
3	36.9	23.0	21.44425	27.70038
4	42.1	24.5	22.52875	28.78824
5	43.4	27.5	22.82119	29.08105
6	46.4	26.0	23.52858	29.78859
7	51.7	27.3	24.88926	31.14709
8	53.0	24.5	25.24465	31.50147
9	53.8	30.7	25.46759	31.72370
10	53.8	28.2	25.46759	31.72370
11	56.2	31.4	26.15578	32.40933
12	60.0	31.0	27.30483	33.55303
13	60.1	32.6	27.33606	33.58409
14	61.1	31.0	27.65105	33.89740
15	61.2	32.5	27.68283	33.92900
16	62.6	30.0	28.13300	34.37661
17	63.3	31.4	28.36179	34.60404
18	63.9	35.1	28.55987	34.80091
19	64.2	33.0	28.65959	34.90001
20	64.2	34.9	28.65959	34.90001
21	66.4	32.2	29.40473	35.64030
22	67.1	30.8	29.64694	35.88086
23	67.6	30.7	29.82147	36.05416
24	67.8	31.5	29.89163	36.12383
25	68.8	35.0	30.24546	36.47513
26	69.6	34.6	30.53216	36.75972
27	69.9	34.9	30.64051	36.86726
28	70.3	31.1	30.78567	37.01133
29	70.7	34.1	30.93165	37.15619
30	70.9	34.9	31.00494	37.22892
31	71.0	35.8	31.04166	37.26535
32	72.3	32.0	31.52361	37.74353
33	72.4	35.4	31.56104	37.78066
34	73.0	36.8	31.78665	38.00446
35	73.1	34.8	31.82443	38.04193
36	73.1	34.4	31.82443	38.04193
37	73.6	35.3	32.01409	38.23004
38	74.1	34.1	32.20500	38.41938
39	74.5	33.2	32.35864	38.57174
40	75.4	36.5	32.70729	38.91744
41	75.6	36.4	32.78532	38.99481
42	76.8	35.4	33.25774	39.46314
43	77.1	36.7	33.37698	39.58133
44	78.2	38.2	33.81808	40.01851
45	78.8	39.4	34.06125	40.25948

	Sa02 (%)	a*	Q2.5	Q97.5
46	79.1	36.9	34.18352	40.38063
47	79.3	36.5	34.26528	40.46165
48	79.8	37.6	34.47058	40.66505
49	80.1	38.5	34.59436	40.78769
50	80.6	37.6	34.80167	40.99306
51	80.7	31.9	34.84328	41.03428
52	80.7	36.3	34.84328	41.03428
53	81.1	37.2	35.01024	41.19967
54	81.2	39.2	35.05210	41.24114
55	81.3	37.5	35.09402	41.28266
56	81.5	40.8	35.17800	41.36585
57	82.0	38.1	35.38884	41.57468
58	82.2	38.4	35.47353	41.65855
59	82.4	38.2	35.55842	41.74263
60	82.8	37.4	35.72881	41.91137
61	83.1	39.4	35.85713	42.03844
62	83.2	35.9	35.90001	42.08090
64	83.6	37.7	36.07201	42.25122
65	83.8	40.4	36.15831	42.33668
66	84.2	39.5	36.33153	42.50818
67	84.4	38.2	36.41843	42.59423
68	84.5	37.7	36.46196	42.63733
69	84.5	42.1	36.46196	42.63733
70	85.1	39.0	36.72420	42.89695
71	85.3	40.7	36.81202	42.98389
72	85.4	37.2	36.85601	43.02743
73	86.1	39.7	37.16531	43.33361
74	86.2	40.1	37.20969	43.37754
75	86.5	39.4	37.34316	43.50965
76	86.6	38.7	37.38775	43.55379
77	86.6	38.4	37.38775	43.55379
78	86.6	40.4	37.38775	43.55379
79	86.7	37.3	37.43239	43.59797
80	86.8	39.7	37.47708	43.64220
81	86.9	41.6	37.52182	43.68648
82	87.5	40.1	37.79132	43.95321
83	87.5	41.0	37.79132	43.95321
84	87.6	41.0	37.83642	43.99784
85	87.8	38.4	37.92676	44.08724
86	88.0	40.5	38.01730	44.17685
87	88.2	40.2	38.10804	44.26664
88	88.3	40.7	38.15349	44.31162
89	88.3	40.8	38.15349	44.31162
90	88.4	40.5	38.19899	44.35664
91	88.8	39.0	38.38148	44.53722
92	88.8	40.6	38.38148	44.53722
93	88.9	41.0	38.42723	44.58249
94	89.0	39.9	38.47303	44.62781
95	89.0	40.7	38.47303	44.62781
96	89.0	41.1	38.47303	44.62781
97	89.1	40.4	38.51888	44.67318
98	89.2	41.4	38.56478	44.71859
99	89.3	40.9	38.61073	44.76406
100	89.6	41.5	38.74889	44.90075
101	89.7	42.5	38.79504	44.94642
102	89.7	39.7	38.79504	44.94642
103	89.9	42.5	38.88750	45.03789

	Sa02 (%)	a*	Q2.5	Q97.5
104	90.0	41.6	38.93381	45.08371
105	90.1	41.0	38.98016	45.12957
106	90.2	41.3	39.02657	45.17548
107	90.3	42.4	39.07302	45.22144
108	90.5	40.0	39.16608	45.31350
109	90.5	39.9	39.16608	45.31350
110	90.6	42.9	39.21269	45.35961
111	90.8	39.4	39.30606	45.45198
112	90.9	40.9	39.35282	45.49823
113	91.2	40.4	39.49339	45.63730
114	91.4	42.4	39.58737	45.73025
115	91.4	41.7	39.58737	45.73025
116	91.4	42.7	39.58737	45.73025
117	91.4	39.9	39.58737	45.73025
118	91.5	43.0	39.63443	45.77680
119	91.9	45.0	39.82318	45.96350
120	92.0	41.2	39.87049	46.01030
121	92.0	42.6	39.87049	46.01030
122	92.0	42.1	39.87049	46.01030
123	92.0	42.5	39.87049	46.01030
124	92.1	42.7	39.91785	46.05715
125	92.2	42.1	39.96527	46.10405
126	92.3	43.0	40.01273	46.15099
127	92.5	41.6	40.10781	46.24503
128	92.5	46.6	40.10781	46.24503
129	92.7	42.7	40.20310	46.33927
130	92.8	40.8	40.25081	46.38647
131	93.5	45.0	40.58624	46.71820
132	93.6	45.2	40.63436	46.76579
133	93.9	46.3	40.77903	46.90885
134	94.3	42.0	40.97262	47.10029
135	94.6	47.8	41.11834	47.24438
136	94.7	46.8	41.16702	47.29251
137	95.0	46.2	41.31335	47.43720
138	95.2	45.1	41.41115	47.53391
139	95.5	45.3	41.55824	47.67933
140	95.5	44.9	41.55824	47.67933
141	95.7	45.8	41.65655	47.77653
142	95.7	45.3	41.65655	47.77653
143	95.7	45.2	41.65655	47.77653
144	95.9	46.2	41.75506	47.87393
145	96.0	44.5	41.80439	47.92270
146	96.2	45.3	41.90320	48.02039
147	96.4	46.3	42.00222	48.11828
148	96.5	44.8	42.05180	48.16730
149	96.5	47.8	42.05180	48.16730
150	96.6	46.2	42.10144	48.21637
151	96.9	45.6	42.25064	48.36387
152	97.1	45.0	42.35036	48.46245
153	97.2	43.8	42.40030	48.51181
154	97.3	46.4	42.45028	48.56123
155	97.5	44.8	42.55041	48.66020
156	97.6	46.2	42.60055	48.70976
157	97.6	45.8	42.60055	48.70976
158	97.6	44.5	42.60055	48.70976
159	98.0	44.6	42.80161	48.90850
160	98.2	46.3	42.90244	49.00817

	Sa02 (%)	a*	Q2.5	Q97.5
161	98.4	46.6	43.00347	49.10803
162	99.0	45.2	43.30778	49.40881
163	99.3	47.6	43.46062	49.55986

```r
# Plot output
ggplot(data = astar_interval) +
    aes(x = Coox_SaO2) +
    geom_point(data = data,
            aes(y = a_star,
                x = Coox_SaO2,
                fill = Animal_ID),
            shape = 21,
            size = 3) +
    geom_smooth(data = astar_interval,
            aes(y = `Q2.5`),
            se = FALSE) +
    geom_smooth(data = astar_interval,
            aes(y = `Q97.5`),
            se = FALSE) +
    scale_fill_viridis_d(name = 'Animal') +
    labs(title = 'Sa02 vs a*',
        subtitle = '95% prediction interval',
        y = 'a*',
        x = 'Sa02 (%)')
```

Sa02 vs a*
95% prediction interval
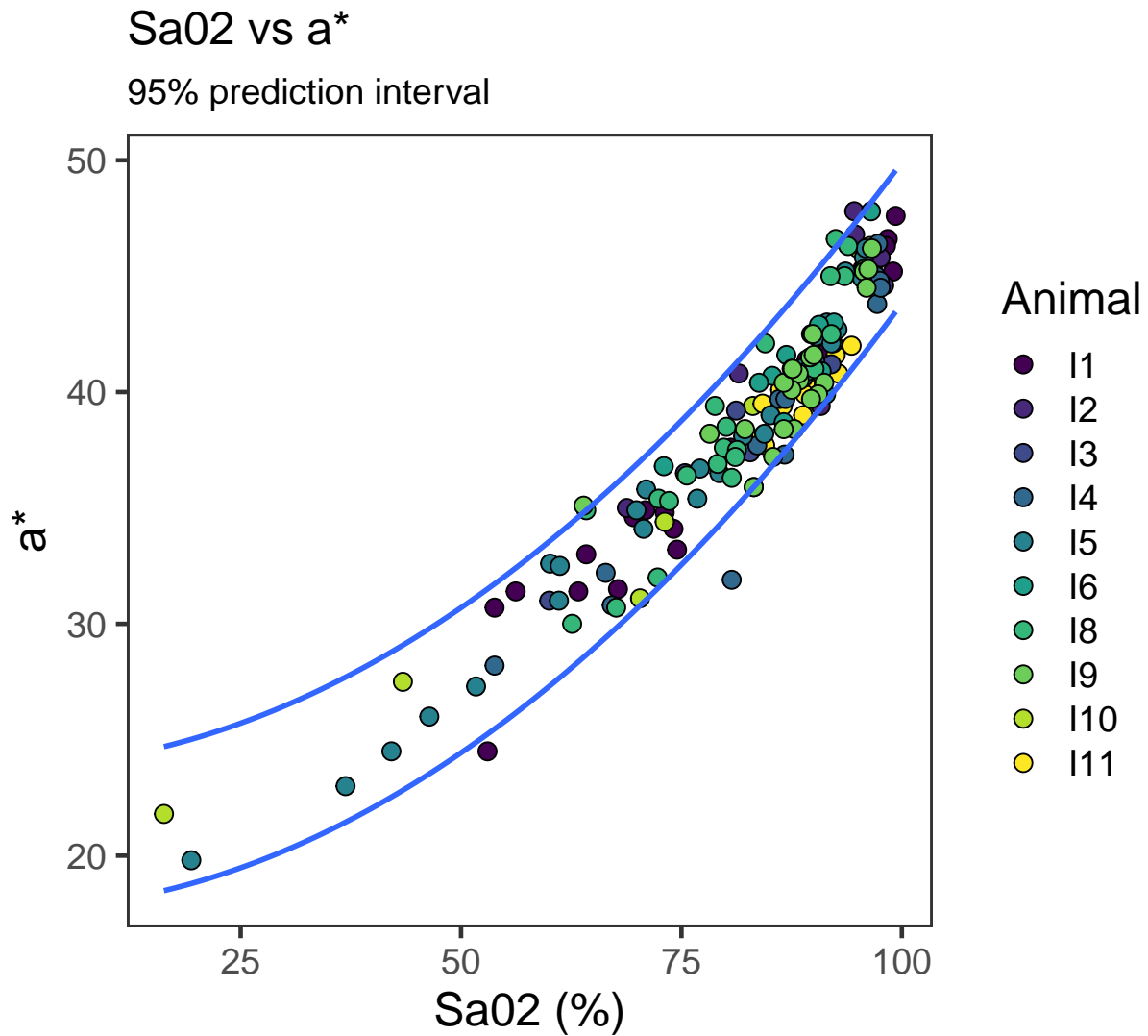
6.5   Generate predicted values

```
# New data
astar_grid <- expand.grid(Coox_SaO2 = c(15, 25, 35, 45, 55, 65, 75, 85, 95),
                          Animal_ID = 'I12')

# Generate predicted vs independent variable SaO2
astar_predicted <- predict(astar_poly,
                           astar_grid,
                           allow.new.levels = TRUE)

# Tabulate
data.frame(SaO2 = astar_grid$Coox_SaO2,
           astar = astar_predicted) %>%
   kable(caption = 'a* predicted values',
         col.names = c('SaO2 (%)', 'a*'))
```

Table 6: a* predicted values

Sa02 (%)	a*
15	20.96506
25	22.10265

Sa02 (%)	a*
35	23.74632
45	25.89607
55	28.55190
65	31.71381
75	35.38180
85	39.55586
95	44.23601

7 b*

7.1 Exploratory plots
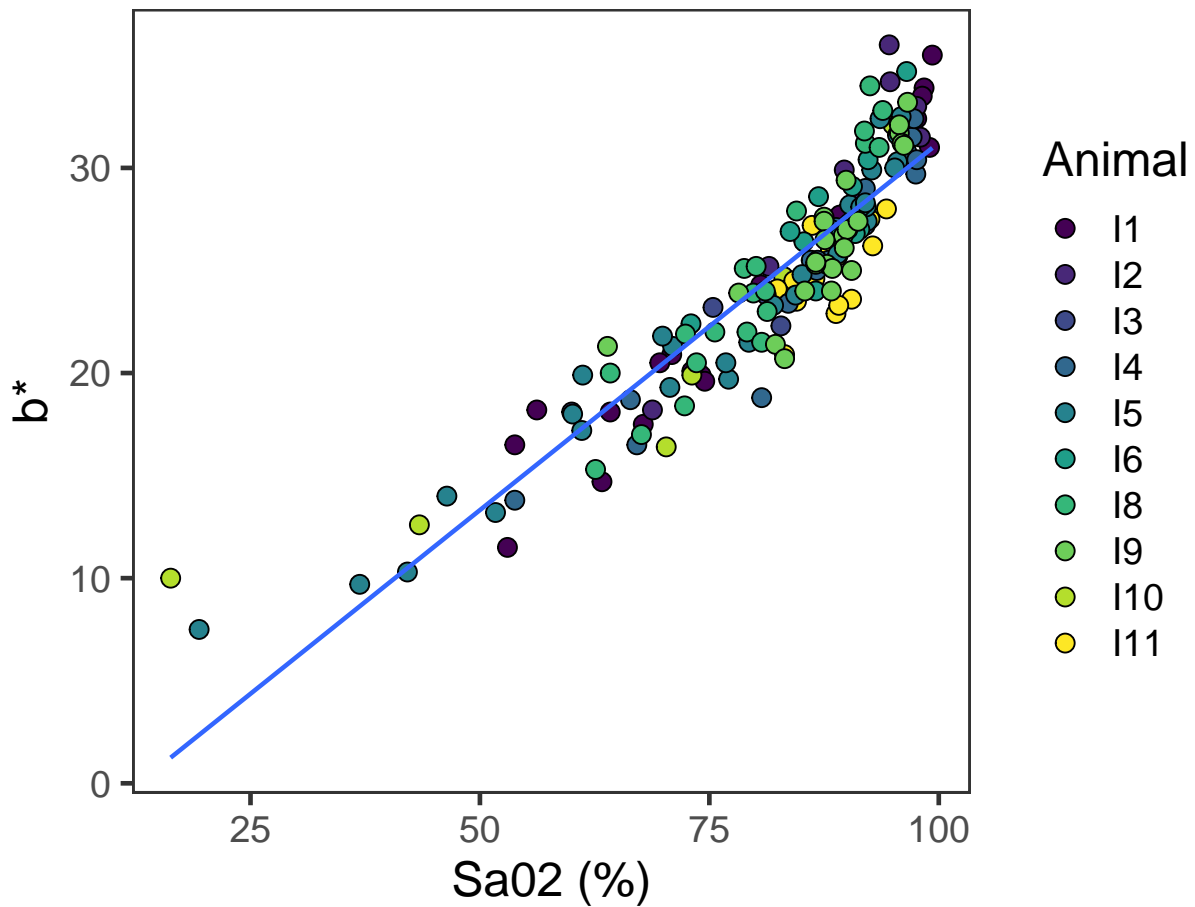
7.1.1 Linear model

```
# Composite plot
ggplot(data = data) +
    aes(y = b_star,
        x = Coox_Sa02,
        fill = Animal_ID) +
    geom_point(shape = 21,
              size = 3) +
    geom_smooth(se = FALSE,
              size = 0.8,
              method = 'lm',
              fill = 'blue') +
    scale_fill_viridis_d(name = 'Animal') +
    labs(title = 'Sa02 vs b*',
        subtitle = 'All animals together\nRegression line is simple linear model',
        x = 'Sa02 (%)',
        y = 'b*')
```

Sa02 vs b*

All animals together
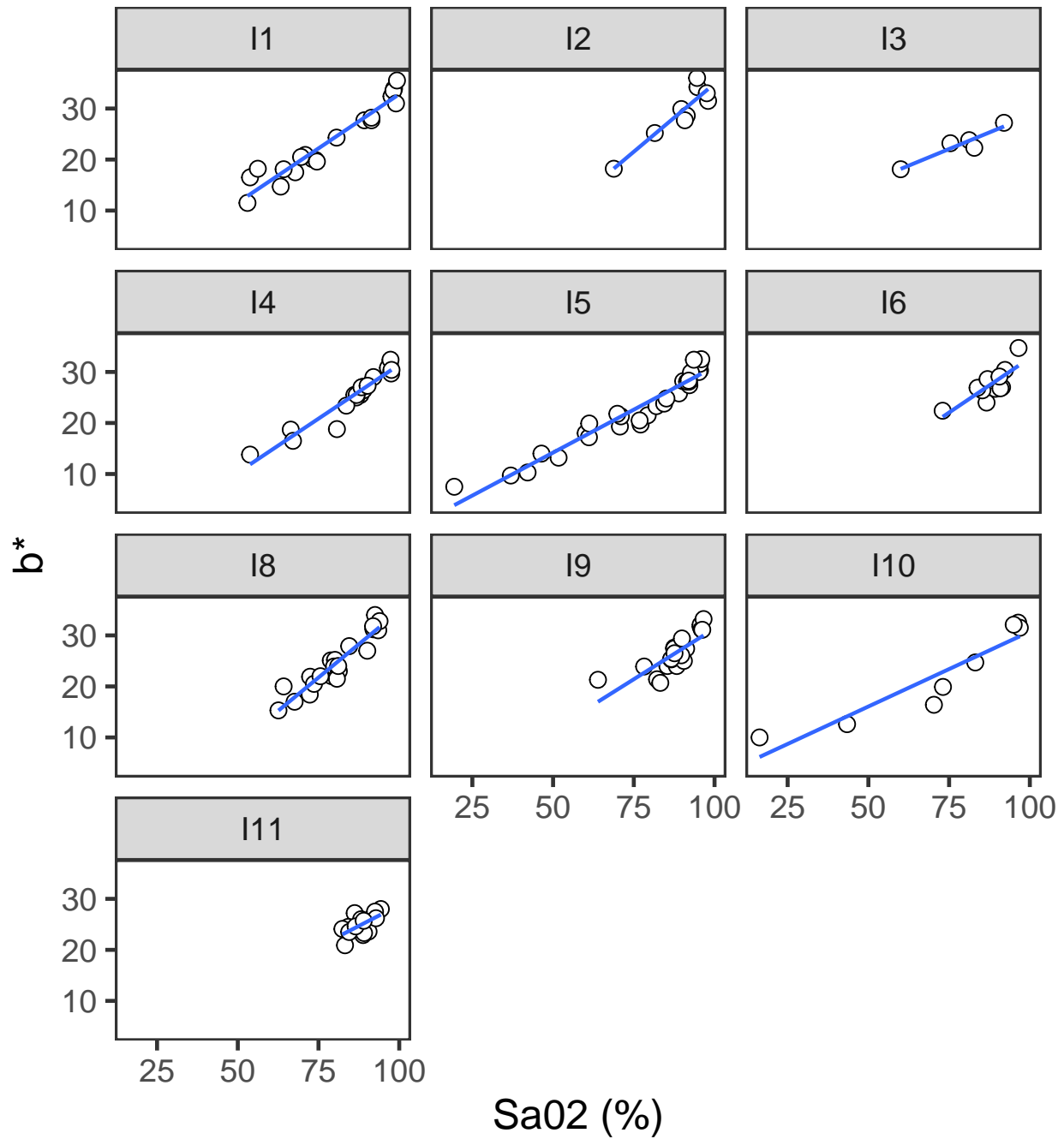Regression line is simple linear model


```r
# Small multiples
ggplot(data = data) +
    aes(y = b_star,
        x = Coox_Sa02) +
    geom_point(shape = 21,
               size = 3,
               fill = '#FFFFFF') +
    geom_smooth(se = FALSE,
                size = 0.8,
                method = 'lm') +
    labs(title = 'Sa02 vs b*',
         subtitle = 'Facetted by animal\nRegression line: linear',
         y = 'b*',
         x = 'Sa02 (%)') +
    facet_wrap(~ Animal_ID,
               ncol = 3)
```

Sa02 vs b*

Facetted by animal
Regression line: linear

7.1.2 Quadratic model
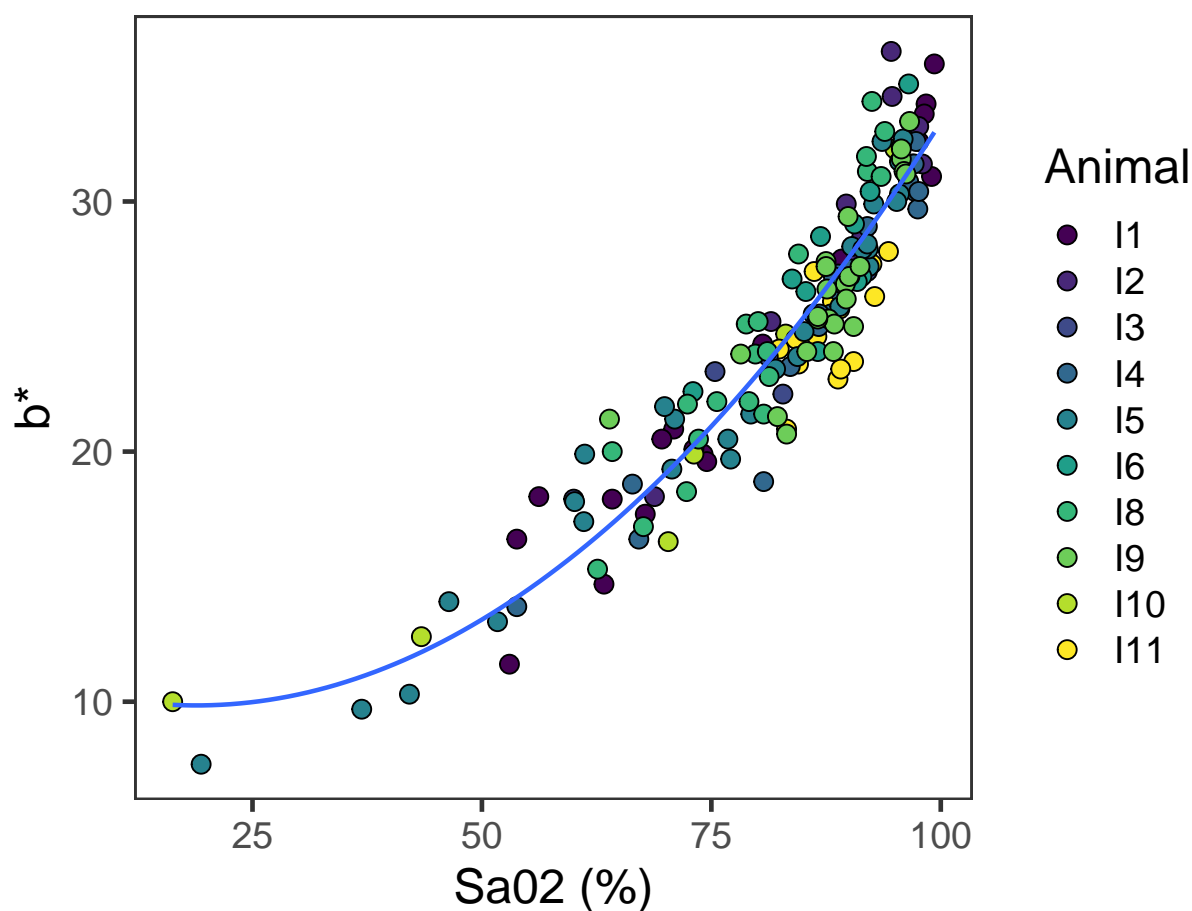
```
# Composite plot
ggplot(data = data) +
    aes(y = b_star,
        x = Coox_SaO2,
        fill = Animal_ID) +
    geom_point(shape = 21,
               size = 3) +
```

```
        geom_smooth(se = FALSE,
                    size = 0.8,
                    method = 'lm',
                    fill = 'blue',
                    formula = y ~ poly(x, 2)) +
        scale_fill_viridis_d(name = 'Animal') +
        labs(title = 'Sa02 vs b*',
             subtitle = 'All animals together\nRegression line is a quadratic polynomial model',
             x = 'Sa02 (%)',
             y = 'b*')
```

Sa02 vs b*

All animals together
Regression line is a quadratic polynomial model


```
# Small multiples
ggplot(data = data) +
    aes(y = b_star,
        x = Coox_Sa02) +
    geom_point(shape = 21,
               size = 3,
               fill = '#FFFFFF') +
    geom_smooth(se = FALSE,
                size = 0.8,
                colour = 'blue',
                method = 'lm',
                formula = y ~ poly(x, 2, raw = TRUE)) +
    labs(title = 'Sa02 vs b*',
```
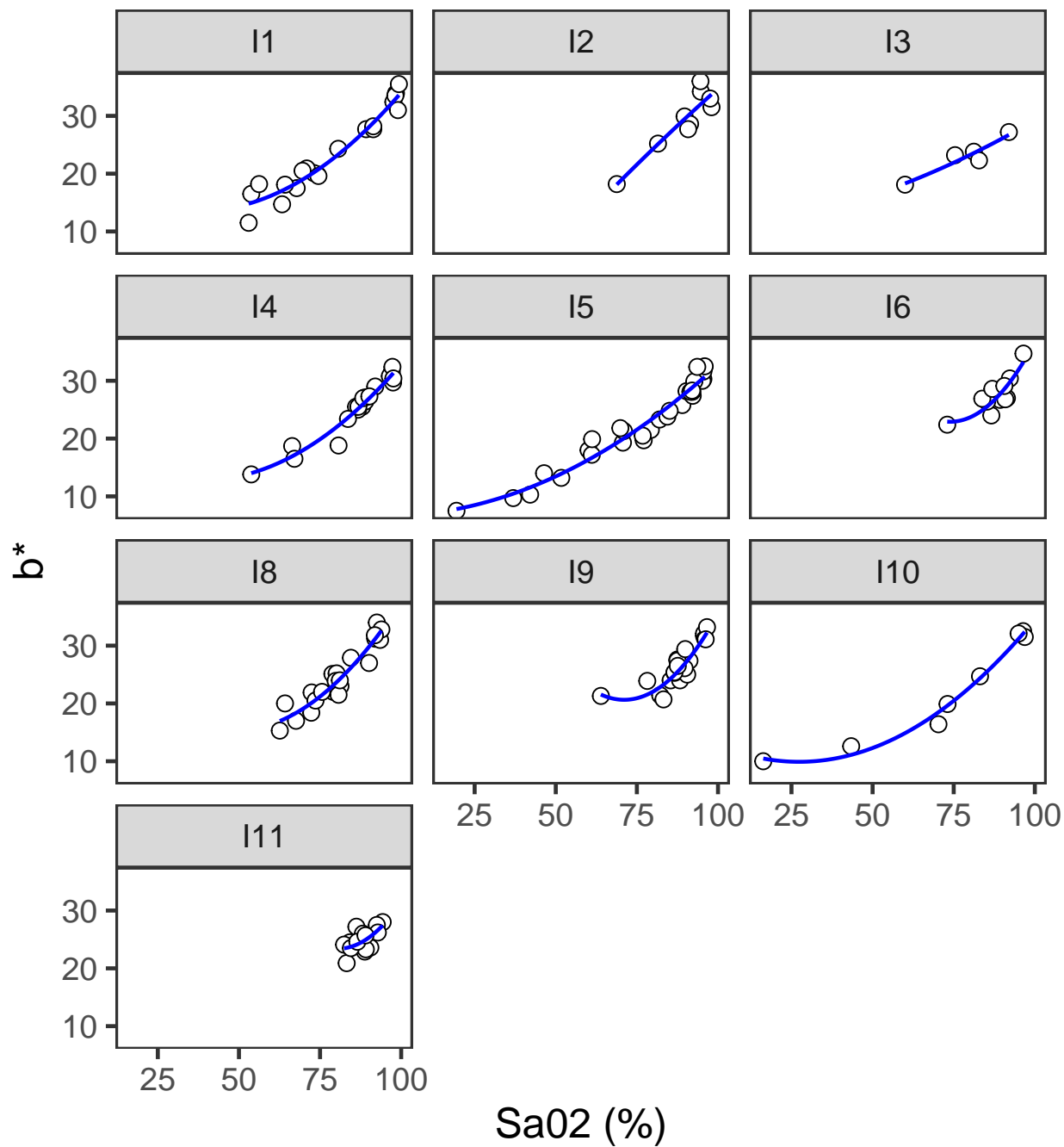
```
        subtitle = 'Facetted by animal\nRegression line: quadratic polynomial',
        y = 'b*',
        x = 'SaO2 (%)') +
    facet_wrap(~ Animal_ID,
               ncol = 3)
```

SaO2 vs b*

Facetted by animal
Regression line: quadratic polynomial

7.2 Model selection

Model selection based on the outcome of likelihood ratio test of goodness of fit.

```r
# Null model
bstar_null <- lmer(b_star ~ 1 + (1 | Animal_ID),
                   data = data,
                   REML = FALSE)

## First-order polynomial (linear) model
bstar_linear <- lmer(b_star ~ Coox_SaO2 + (1 | Animal_ID),
                   data = data,
                   REML = FALSE)

## Second-order polynomial (quadratic) model
bstar_poly <- lmer(b_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID),
                   data = data,
                   REML = FALSE)

# Compare model fits
anova(bstar_null, bstar_linear, bstar_poly, test = 'LRT')
```

```
## Data: data
## Models:
## bstar_null: b_star ~ 1 + (1 | Animal_ID)
## bstar_linear: b_star ~ Coox_SaO2 + (1 | Animal_ID)
## bstar_poly: b_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID)
##              Df     AIC     BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## bstar_null    3 1036.44 1045.72 -515.22  1030.44
## bstar_linear  4  730.04  742.42 -361.02   722.04 308.397      1  < 2.2e-16 ***
## bstar_poly    5  659.79  675.26 -324.90   649.79  72.252      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Best fit model is the second-order polynomial (quadratic) model.

## 7.3 Explore the best fit model

```r
# Summary output
summary(bstar_poly)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: b_star ~ poly(Coox_SaO2, 2) + (1 | Animal_ID)
##    Data: data
##
##       AIC      BIC   logLik deviance df.resid
##     659.8    675.3   -324.9    649.8      158
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.28500 -0.73537 -0.02434  0.70548  2.80687
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Animal_ID (Intercept) 0.6551   0.8094
##  Residual              2.8811   1.6974
## Number of obs: 163, groups:  Animal_ID, 10
##
## Fixed effects:
##                     Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)          25.0024     0.2953   9.6667  84.681 3.33e-15 ***
```

```
## poly(Coox_SaO2, 2)1  68.5032     1.7981 161.0661   38.097  < 2e-16 ***
## poly(Coox_SaO2, 2)2  16.8702     1.7627 159.9209    9.571  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) p(C_SO2,2)1
## p(C_SO2,2)1 -0.005
## p(C_SO2,2)2 -0.008 -0.014
```

```
# Check model assumptions
plot_model(model = bstar_poly,
           type = 'diag')[2:4]
```
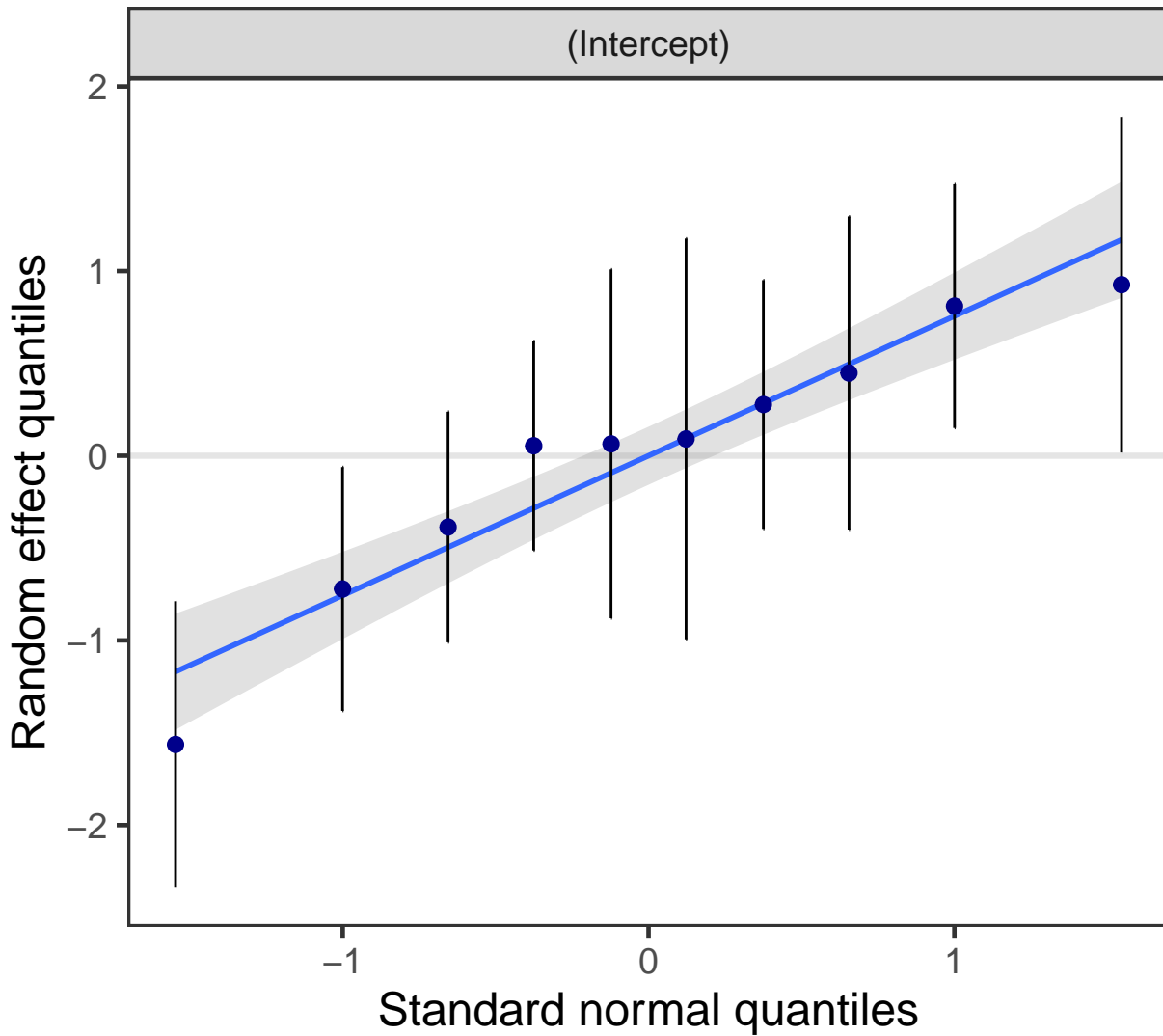
```
## [[1]]
## [[1]]$Animal_ID
```



```
##
##
## [[2]]
```

Non−normality of residuals

Distribution should look like normal curve


```
## 
## [[3]]
```
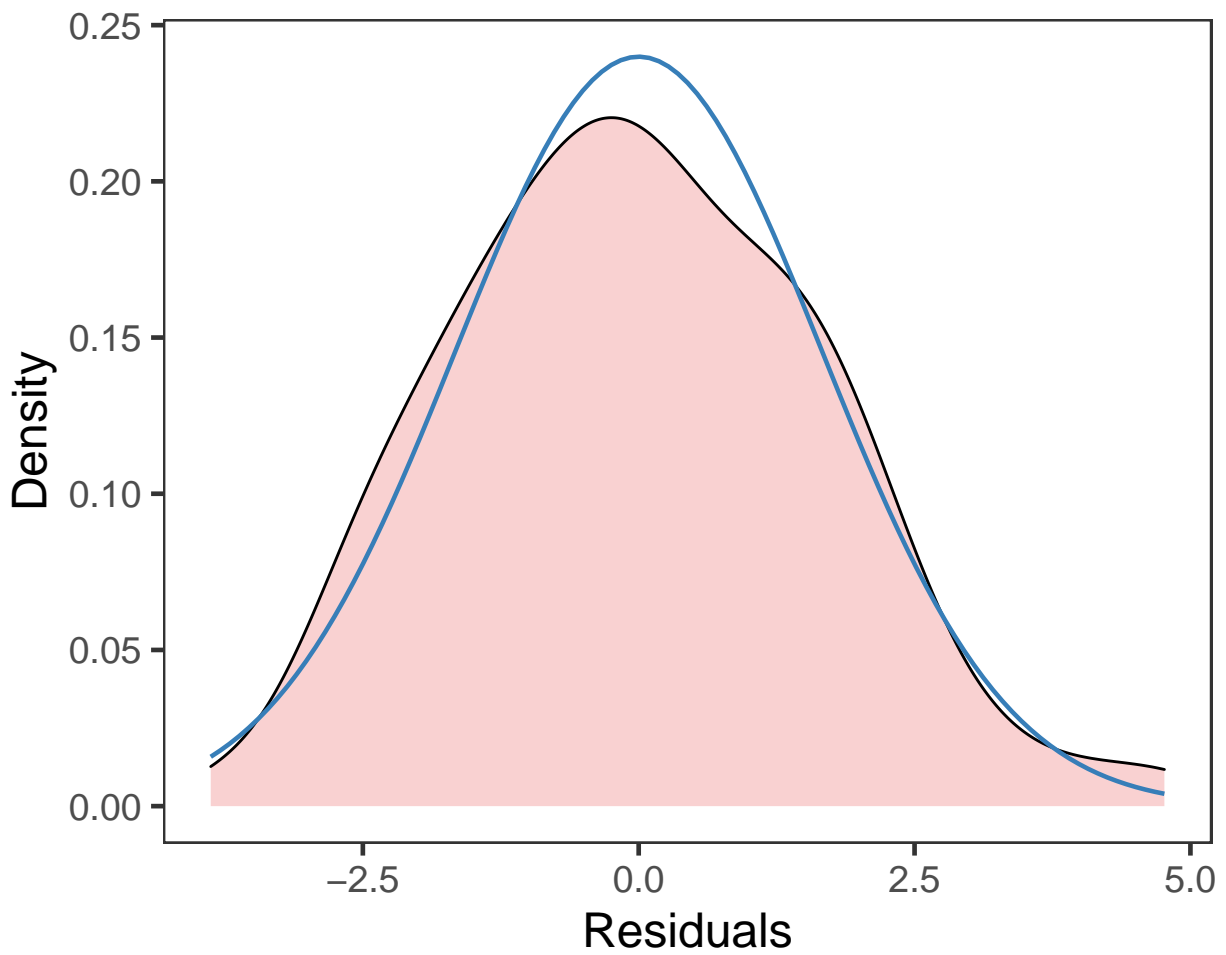
Homoscedasticity (constant variance of residuals)

Amount and distance of points scattered above/below line is equa
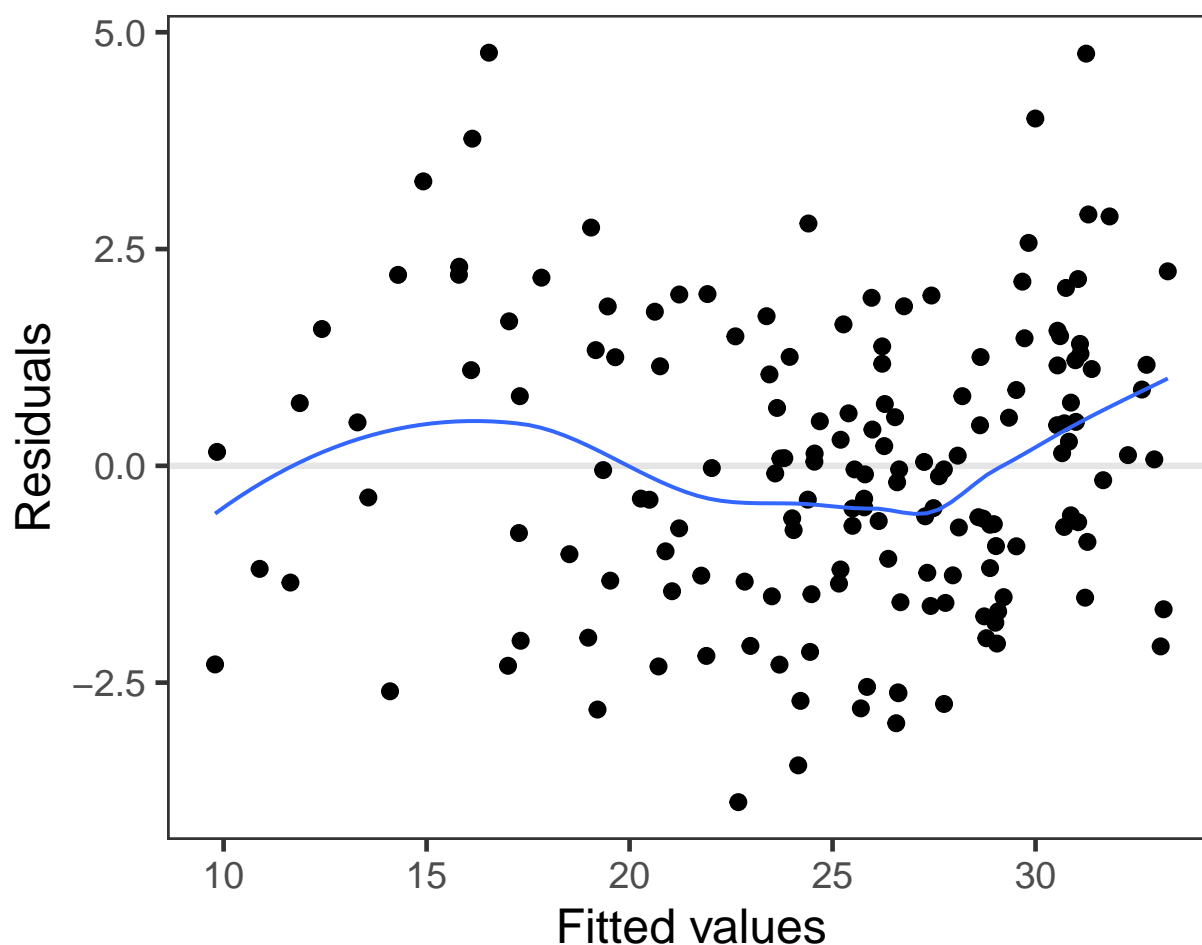

```r
# R-squared (based on likelihood ratio test)
r.squaredLR(bstar_poly, null = bstar_null)
```

```
## [1] 0.9032156
## attr(,"adj.r.squared")
## [1] 0.9048415
```

```r
# Root mean square error
RMSE.merMod(bstar_poly)
```

```
## [1] 1.657782
```

7.4 Generate prediction intervals

```r
# Prediction interval
bstar_lqmm <- lqmm(fixed = b_star ~ poly(Coox_SaO2, 2),
                   random = ~ 1,
                   group = ~ Animal_ID,
                   tau = c(0.025, 0.976),
                   data = data)

bstar_interval <- as.data.frame(predict(bstar_lqmm, level = 0))

names(bstar_interval) <- paste0('Q', c(2.5, 97.5))
```

```
bstar_interval <- cbind(data, bstar_interval)

# Numeric output
bstar_interval %>%
    dplyr::select(Coox_SaO2, b_star, Q2.5, Q97.5) %>%
    arrange(Coox_SaO2) %>%
    unique(.) %>%
    kable(caption = 'b*: 95% prediction interval',
          col.names = c('SaO2 (%)', 'b*', 'Q2.5', 'Q97.5'))
```

Table 7: b*: 95% prediction interval

SaO2 (%)	b*	Q2.5	Q97.5
16.3	10.0	6.406708	13.83413
19.4	7.5	6.349370	13.79463
36.9	9.7	7.337431	14.87631
42.1	10.3	8.060536	15.62490
43.4	12.6	8.272059	15.84263
46.4	14.0	8.807128	16.39177
51.7	13.2	9.912483	17.52112
53.0	11.5	10.214828	17.82918
53.8	16.5	10.406999	18.02483
53.8	13.8	10.406999	18.02483
56.2	18.2	11.011458	18.63959
60.0	18.1	12.054244	19.69822
60.1	18.0	12.083105	19.72749
61.1	17.2	12.375716	20.02417
61.2	19.9	12.405378	20.05424
62.6	15.3	12.828278	20.48277
63.3	14.7	13.045077	20.70235
63.9	21.3	13.233743	20.89339
64.2	18.1	13.329058	20.98989
64.2	20.0	13.329058	20.98989
66.4	18.7	14.048050	21.71743
67.1	16.5	14.284207	21.95627
67.6	17.0	14.455074	22.12904
67.8	17.5	14.523930	22.19865
68.8	18.2	14.872576	22.55106
69.6	20.5	15.156733	22.83821
69.9	21.8	15.264493	22.94708
70.3	16.4	15.409191	23.09326
70.7	19.3	15.555054	23.24059
70.9	20.9	15.628422	23.31470
71.0	21.3	15.665215	23.35185
72.3	18.4	16.150149	23.84152
72.4	21.9	16.187961	23.87969
73.0	22.4	16.416361	24.11025
73.1	20.1	16.454683	24.14892
73.1	19.9	16.454683	24.14892
73.6	20.5	16.647381	24.34341
74.1	19.9	16.841899	24.53970
74.5	19.6	16.998824	24.69804
75.4	23.2	17.356161	25.05853
75.6	22.0	17.436369	25.13944
76.8	20.5	17.923735	25.63095
77.1	19.7	18.047214	25.75546
78.2	23.9	18.505573	26.21756
78.8	25.1	18.759298	26.47330

Sa02 (%)	b*	Q2.5	Q97.5
79.1	22.0	18.887143	26.60215
79.3	21.5	18.972737	26.68841
79.8	23.9	19.187996	26.90533
80.1	25.2	19.318024	27.03635
80.6	24.3	19.536194	27.25617
80.7	18.8	19.580046	27.30035
80.7	21.5	19.580046	27.30035
81.1	24.0	19.756182	27.47780
81.2	23.8	19.800398	27.52234
81.3	23.0	19.844687	27.56695
81.5	25.2	19.933483	27.65640
82.0	23.3	20.156747	27.88128
82.2	21.4	20.246562	27.97174
82.4	24.1	20.336668	28.06249
82.8	22.3	20.517753	28.24486
83.1	24.7	20.654331	28.38239
83.2	20.9	20.700003	28.42838
83.2	20.7	20.700003	28.42838
83.6	23.4	20.883416	28.61307
83.8	26.9	20.975560	28.70584
84.2	24.5	21.160721	28.89226
84.4	23.8	21.253737	28.98591
84.5	23.5	21.300355	29.03284
84.5	27.9	21.300355	29.03284
85.1	24.8	21.581589	29.31594
85.3	26.4	21.675916	29.41089
85.4	24.0	21.723188	29.45847
86.1	25.5	22.056134	29.79358
86.2	27.2	22.103989	29.84174
86.5	24.6	22.247990	29.98666
86.6	24.0	22.296135	30.03511
86.6	25.3	22.296135	30.03511
86.6	25.4	22.296135	30.03511
86.7	25.0	22.344354	30.08363
86.8	25.5	22.392646	30.13222
86.9	28.6	22.441010	30.18089
87.5	27.6	22.732723	30.47442
87.5	27.4	22.732723	30.47442
87.6	26.5	22.781597	30.52359
87.8	25.3	22.879562	30.62216
88.0	25.5	22.977819	30.72102
88.2	26.0	23.076366	30.82016
88.3	27.0	23.125749	30.86984
88.3	24.0	23.125749	30.86984
88.4	25.1	23.175205	30.91960
88.8	22.9	23.373756	31.11934
88.8	27.1	23.373756	31.11934
88.9	26.4	23.423576	31.16945
89.0	25.7	23.473468	31.21964
89.0	26.6	23.473468	31.21964
89.0	25.8	23.473468	31.21964
89.1	23.3	23.523433	31.26990
89.2	27.7	23.573471	31.32023
89.3	26.7	23.623582	31.37064
89.6	26.7	23.774351	31.52229
89.7	29.9	23.824753	31.57298
89.7	26.1	23.824753	31.57298

Sa02 (%)	b*	Q2.5	Q97.5
89.9	29.4	23.925775	31.67459
90.0	27.0	23.976395	31.72550
90.1	27.0	24.027088	31.77648
90.2	27.3	24.077853	31.82754
90.3	28.2	24.128692	31.87867
90.5	23.6	24.230587	31.98114
90.5	25.0	24.230587	31.98114
90.6	29.1	24.281644	32.03249
90.8	27.7	24.383976	32.13540
90.9	26.8	24.435251	32.18696
91.2	27.4	24.589513	32.34208
91.4	27.7	24.692718	32.44586
91.4	28.2	24.692718	32.44586
91.4	28.6	24.692718	32.44586
91.4	27.0	24.692718	32.44586
91.5	28.1	24.744430	32.49786
91.9	31.8	24.952005	32.70657
92.0	27.2	25.004080	32.75893
92.0	29.0	25.004080	32.75893
92.0	28.3	25.004080	32.75893
92.0	31.2	25.004080	32.75893
92.1	28.1	25.056229	32.81136
92.2	27.4	25.108450	32.86387
92.3	30.4	25.160744	32.91644
92.5	27.5	25.265550	33.02181
92.5	34.0	25.265550	33.02181
92.7	29.9	25.370647	33.12747
92.8	26.2	25.423305	33.18041
93.5	31.0	25.793948	33.55301
93.6	32.4	25.847188	33.60652
93.9	32.8	26.007344	33.76751
94.3	28.0	26.221905	33.98317
94.6	36.0	26.383590	34.14568
94.7	34.2	26.437631	34.19999
95.0	32.1	26.600189	34.36337
95.2	30.0	26.708925	34.47265
95.5	30.3	26.872575	34.63711
95.5	31.6	26.872575	34.63711
95.7	32.2	26.982039	34.74711
95.7	31.7	26.982039	34.74711
95.7	32.1	26.982039	34.74711
95.9	32.5	27.091793	34.85740
96.0	31.2	27.146780	34.91266
96.2	31.1	27.256971	35.02338
96.4	32.5	27.367454	35.13440
96.5	30.8	27.422804	35.19001
96.5	34.7	27.422804	35.19001
96.6	33.2	27.478228	35.24570
96.9	31.5	27.644934	35.41320
97.1	31.5	27.756435	35.52523
97.2	30.4	27.812295	35.58136
97.3	32.4	27.868228	35.63755
97.5	29.7	27.980311	35.75016
97.6	32.4	28.036462	35.80657
97.6	33.0	28.036462	35.80657
97.6	30.4	28.036462	35.80657
98.0	31.5	28.261794	36.03295

Sa02 (%)	b*	Q2.5	Q97.5
98.2	33.5	28.374896	36.14657
98.4	33.9	28.488290	36.26048
99.0	31.0	28.830217	36.60395
99.3	35.5	29.002163	36.77666

```r
# Plot output
ggplot(data = bstar_interval) +
    aes(x = Coox_Sa02) +
    geom_point(data = data,
               aes(y = b_star,
                   x = Coox_Sa02,
                   fill = Animal_ID),
               shape = 21,
               size = 3) +
    geom_smooth(data = bstar_interval,
                aes(y = `Q2.5`),
                se = FALSE) +
    geom_smooth(data = bstar_interval,
                aes(y = `Q97.5`),
                se = FALSE) +
    scale_fill_viridis_d(name = 'Animal') +
    labs(title = 'Sa02 vs b*',
         subtitle = '95% prediction interval',
         y = 'b*',
         x = 'Sa02 (%)')
```

Sa02 vs b*
95% prediction interval
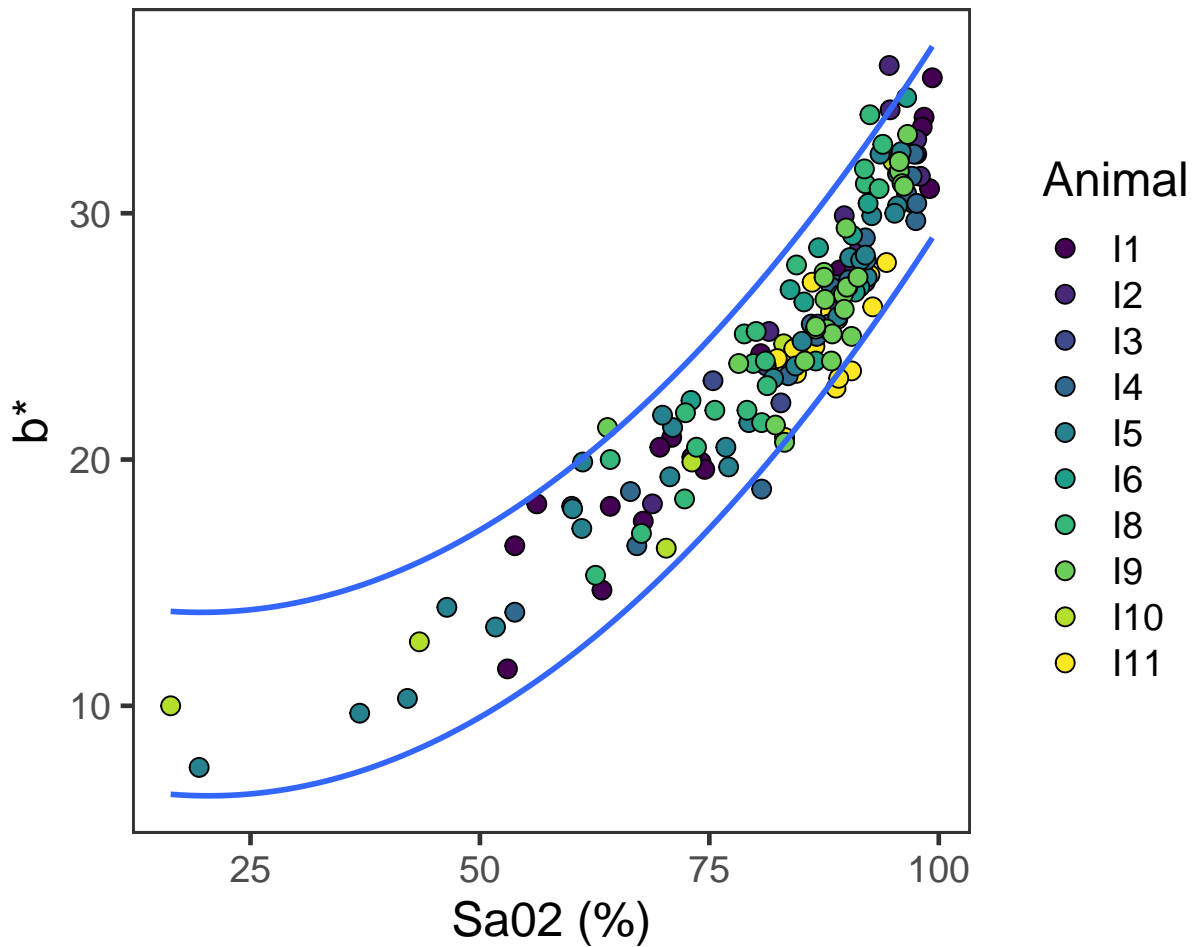
7.5 Generate predicted values

```r
# New data
bstar_grid <- expand.grid(Coox_SaO2 = c(15, 25, 35, 45, 55, 65, 75, 85, 95),
                          Animal_ID = 'I12')

# Generate predicted vs independent variable SaO2
bstar_predicted <- predict(bstar_poly,
                           bstar_grid,
                           allow.new.levels = TRUE)

# Tabulate
data.frame(SaO2 = bstar_grid$Coox_SaO2,
           bstar = bstar_predicted) %>%
   kable(caption = 'b* predicted values',
         col.names = c('SaO2 (%)', 'b*'))
```

Table 8: b* predicted values

Sa02 (%)	b*
15	9.812421
25	9.843973

44

Sa02 (%)	b*
35	10.606793
45	12.100880
55	14.326235
65	17.282858
75	20.970748
85	25.389906
95	30.540331

8 Code for publication plots

```r
# Individual plots
l <- ggplot(data = lstar_interval) +
    aes(x = Coox_Sa02) +
    geom_smooth(data = data,
                aes(y = l_star),
                se = FALSE,
                colour = '#000000',
                method = 'lm',
                formula = y ~ poly(x, 2, raw = TRUE)) +
    geom_smooth(data = lstar_interval,
                aes(y = `Q2.5`),
                se = FALSE) +
    geom_smooth(data = lstar_interval,
                aes(y = `Q97.5`),
                se = FALSE) +
    annotate(geom = 'text',
             x = 29, y = 24.2,
             label = 'Pseudo-R ^2 == 0.94',
             parse = TRUE,
             size = 4) +
    annotate(geom = 'text',
             x = 25, y = 23.2,
             label = 'RMSE = 0.66',
             size = 4) +
    scale_y_continuous(limits = c(8, 25),
                       breaks = c(12, 16, 20, 24),
                       expand = c(0, 0)) +
    labs(y = 'L*',
         x = expression('SaO'[2]*' (%)'))

ggsave(filename = 'figures/sao2-L.png',
       plot = l,
       height = 4,
       width = 5)

a <- ggplot(data = astar_interval) +
    aes(x = Coox_Sa02) +
    geom_smooth(data = data,
                aes(y = a_star),
                se = FALSE,
                colour = '#000000',
                method = 'lm',
```

```r
                    formula = y ~ poly(x, 2, raw = TRUE)) +
    geom_smooth(data = astar_interval,
                aes(y = `Q2.5`),
                se = FALSE) +
    geom_smooth(data = astar_interval,
                aes(y = `Q97.5`),
                se = FALSE) +
    annotate(geom = 'text',
             x = 29, y = 52,
             label = 'Pseudo-R ^2 == 0.93',
             parse = TRUE,
             size = 4) +
    annotate(geom = 'text',
             x = 25, y = 50,
             label = 'RMSE = 1.42',
             size = 4) +
    scale_y_continuous(limits = c(18, 52),
                       breaks = c(20, 30, 40, 50)) +
    labs(y = 'a*',
         x = expression('SaO'[2]*' (%)'))

ggsave(filename = 'figures/sao2-a.png',
       plot = a,
       height = 4,
       width = 5)

b <- ggplot(data = bstar_interval) +
    aes(x = Coox_SaO2) +
    geom_smooth(data = data,
                aes(y = b_star),
                se = FALSE,
                colour = '#000000',
                method = 'lm',
                formula = y ~ poly(x, 2, raw = TRUE)) +
    geom_smooth(data = bstar_interval,
                aes(y = `Q2.5`),
                se = FALSE) +
    geom_smooth(data = bstar_interval,
                aes(y = `Q97.5`),
                se = FALSE) +
    annotate(geom = 'text',
             x = 28, y = 39,
             label = 'Pseudo-R ^2 == 0.90',
             parse = TRUE,
             size = 4) +
    annotate(geom = 'text',
             x = 25, y = 37,
             label = 'RMSE = 1.66',
             size = 4) +
    scale_y_continuous(limits = c(6, 39),
                       breaks = c(10, 20, 30)) +
    labs(y = 'b*',
         x = expression('SaO'[2]*' (%)'))

ggsave(filename = 'figures/sao2-b.png',
       plot = b,
       height = 4,
       width = 5)
```

```r
# Stacked plot
lplot <- ggplot(data = lstar_interval) +
    aes(x = Coox_SaO2) +
    geom_smooth(data = data,
                aes(y = l_star),
                se = FALSE,
                colour = '#000000',
                method = 'lm',
                formula = y ~ poly(x, 2, raw = TRUE)) +
    geom_smooth(data = lstar_interval,
                aes(y = `Q2.5`),
                se = FALSE) +
    geom_smooth(data = lstar_interval,
                aes(y = `Q97.5`),
                se = FALSE) +
    labs(y = 'L*') +
    theme(axis.title.x = element_blank())

aplot <- ggplot(data = astar_interval) +
    aes(x = Coox_SaO2) +
    geom_smooth(data = data,
                aes(y = a_star),
                se = FALSE,
                colour = '#000000',
                method = 'lm',
                formula = y ~ poly(x, 2, raw = TRUE)) +
    geom_smooth(data = astar_interval,
                aes(y = `Q2.5`),
                se = FALSE) +
    geom_smooth(data = astar_interval,
                aes(y = `Q97.5`),
                se = FALSE) +
    labs(y = 'a*') +
    theme(axis.title.x = element_blank(),
          legend.position = 'none')

bplot <- ggplot(data = bstar_interval) +
    aes(x = Coox_SaO2) +
    geom_smooth(data = data,
                aes(y = b_star),
                se = FALSE,
                colour = '#000000',
                method = 'lm',
                formula = y ~ poly(x, 2, raw = TRUE)) +
    geom_smooth(data = bstar_interval,
                aes(y = `Q2.5`),
                se = FALSE) +
    geom_smooth(data = bstar_interval,
                aes(y = `Q97.5`),
                se = FALSE) +
    labs(y = 'b*',
         x = expression('SaO'[2]*' (%)')) +
    theme(legend.position = 'none')

# Construct plot
pub_plot <- lplot + aplot + bplot +
    plot_layout(ncol = 1) +
    plot_annotation(tag_levels = 'A')
```

```
ggsave(filename = 'figures/figure_1a.png',
       plot = pub_plot,
       height = 11,
       width = 6)
```

---

9 Session Information

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] patchwork_1.0.0    knitr_1.28         MuMIn_1.43.15      lqmm_1.5.5
##  [5] sjPlot_2.8.3       merTools_0.5.0     arm_1.10-1         MASS_7.3-51.5
##  [9] lmerTest_3.1-1     lme4_1.1-21        Matrix_1.2-18      skimr_2.1
## [13] magrittr_1.5       forcats_0.5.0      stringr_1.4.0      dplyr_0.8.5
## [17] purrr_0.3.3        readr_1.3.1        tidyr_1.0.2        tibble_3.0.0
## [21] ggplot2_3.3.0.9000 tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] TH.data_1.0-10    minqa_1.2.4       colorspace_1.4-1
##  [4] ellipsis_0.3.0    sjlabelled_1.1.3  estimability_1.3
##  [7] parameters_0.6.0  base64enc_0.1-3   fs_1.3.1
## [10] rstudioapi_0.11   glmmTMB_1.0.1     farver_2.0.3
## [13] fansi_0.4.1       mvtnorm_1.1-0     lubridate_1.7.4
## [16] xml2_1.3.0        codetools_0.2-16  splines_3.6.3
## [19] sjmisc_2.8.4      jsonlite_1.6.1    nloptr_1.2.2.1
## [22] ggeffects_0.14.2  broom_0.5.5       dbplyr_1.4.2
## [25] effectsize_0.2.0  shiny_1.4.0.2     compiler_3.6.3
## [28] httr_1.4.1        sjstats_0.17.9    emmeans_1.4.5
## [31] backports_1.1.5   assertthat_0.2.1  fastmap_1.0.1
## [34] cli_2.0.2         later_1.0.0       htmltools_0.4.0
## [37] tools_3.6.3       SparseGrid_0.8.2  coda_0.19-3
## [40] gtable_0.3.0      glue_1.3.2        Rcpp_1.0.4
## [43] cellranger_1.1.0  vctrs_0.2.4       nlme_3.1-145
## [46] iterators_1.0.12  insight_0.8.2     xfun_0.12
## [49] rvest_0.3.5       mime_0.9          lifecycle_0.2.0
## [52] zoo_1.8-7         scales_1.1.0      hms_0.5.3
## [55] promises_1.1.0    sandwich_2.5-1    TMB_1.7.16
## [58] RColorBrewer_1.1-2 yaml_2.2.1       stringi_1.4.6
## [61] highr_0.8         bayestestR_0.5.3  foreach_1.5.0
## [64] blme_1.0-4        boot_1.3-24       repr_1.1.0
```

```
## [67] rlang_0.4.5          pkgconfig_2.0.3   evaluate_0.14
## [70] lattice_0.20-38      labeling_0.3      tidyselect_1.0.0
## [73] R6_2.4.1             generics_0.0.2    multcomp_1.4-12
## [76] DBI_1.1.0            mgcv_1.8-31       pillar_1.4.3
## [79] haven_2.2.0          withr_2.1.2       survival_3.1-11
## [82] abind_1.4-5          performance_0.4.5 modelr_0.1.6
## [85] crayon_1.3.4         utf8_1.1.4        rmarkdown_2.1
## [88] grid_3.6.3           readxl_1.3.1      reprex_0.3.0
## [91] digest_0.6.25        xtable_1.8-4      httpuv_1.5.2
## [94] numDeriv_2016.8-1.1 stats4_3.6.3       munsell_0.5.0
## [97] viridisLite_0.3.0
```