

MULTIPLE LINEAR REGRESSION

EXAMPLE: BLOOD VISCOSITY AND PACKED CELL VOLUME

The following blood viscosity data studied earlier are a good example of where multiple regression could be used. Recall that the data blood viscosity in samples taken from 32 hospital patients. We wish to model viscosity (y) as a function three covariates

- Packed Cell Volume (PCV), x_1 .
- Plasma Fibrinogen, x_2 .
- Plasma Protein, x_3 .

Unit	Viscosity y	PCV x_1	Plasma Fib. x_2	Plasma Pro. x_3
1	3.71	40.00	344	6.27
2	3.78	40.00	330	4.86
3	3.85	42.50	280	5.09
4	3.88	42.00	418	6.79
5	3.98	45.00	774	6.40
6	4.03	42.00	388	5.48
7	4.05	42.50	336	6.27
8	4.14	47.00	431	6.89
9	4.14	46.75	276	5.18
10	4.20	48.00	422	5.73
11	4.20	46.00	280	5.89
12	4.27	47.00	460	6.58
13	4.27	43.25	412	5.67
14	4.37	45.00	320	6.23
15	4.41	50.00	502	4.99
16	4.64	45.00	550	6.37
17	4.68	51.25	414	6.40
18	4.73	50.25	304	6.00
19	4.87	49.00	472	5.94
20	4.94	50.00	728	5.16
21	4.95	50.00	716	6.29
22	4.96	49.00	400	5.96
23	5.02	50.50	576	5.90
24	5.02	51.25	354	5.81
25	5.12	49.50	392	5.49
26	5.15	56.00	352	5.41
27	5.17	50.00	572	6.24
28	5.18	47.00	634	6.50
29	5.38	53.25	458	6.60
30	5.77	57.00	1070	4.82
31	5.90	54.00	488	5.70
32	5.90	54.00	488	5.70

We consider four analyses:

Multiple regression : $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$

Regression on x_1 : $Y = \beta_0 + \beta_1 x_1 + \epsilon$

Regression on x_2 : $Y = \beta_0 + \beta_2 x_2 + \epsilon$

Regression on x_3 : $Y = \beta_0 + \beta_3 x_3 + \epsilon$

```
> mv.visc<-lm(viscosity~pcv+plasfib+plaspro,data=Viscosity)
> summary(mv.visc)

+
+ Call:
+ lm(formula = viscosity ~ pcv + plasfib + plaspro, data = Viscosity)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -0.4661 -0.1734  0.0285  0.1693  0.5525
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept) -1.378238   0.896665  -1.54    0.14
+ pcv          0.116823   0.013609   8.58 2.5e-09 ***
+ plasfib      0.000402   0.000351   1.15    0.26
+ plaspro      0.040036   0.097153   0.41    0.68
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 0.304 on 28 degrees of freedom
+ Multiple R-squared:  0.784, Adjusted R-squared:  0.761
+ F-statistic: 33.9 on 3 and 28 DF, p-value: 1.88e-09

> m1.visc<-lm(viscosity~pcv,data=Viscosity)
> summary(m1.visc)

+
+ Call:
+ lm(formula = viscosity ~ pcv, data = Viscosity)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -0.4881 -0.2215  0.0469  0.1296  0.6492
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept) -1.2234     0.5842   -2.09   0.045 *
+ pcv          0.1224     0.0121  10.09 3.7e-11 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 0.301 on 30 degrees of freedom
+ Multiple R-squared:  0.772, Adjusted R-squared:  0.765
+ F-statistic: 102 on 1 and 30 DF, p-value: 3.73e-11
```

```

> m2.visc<-lm(viscosity~plasfib,data=Viscosity)
> summary(m2.visc)

+
+ Call:
+ lm(formula = viscosity ~ plasfib, data = Viscosity)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -1.175 -0.373 -0.122  0.351  1.219
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)  3.870813   0.292450   13.24  4.6e-14 ***
+ plasfib       0.001659   0.000589    2.82  0.0085 **
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 0.561 on 30 degrees of freedom
+ Multiple R-squared:  0.209, Adjusted R-squared:  0.183
+ F-statistic: 7.93 on 1 and 30 DF, p-value: 0.0085

> m3.visc<-lm(viscosity~plaspro,data=Viscosity)
> summary(m3.visc)

+
+ Call:
+ lm(formula = viscosity ~ plaspro, data = Viscosity)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -0.9797 -0.4863  0.0686  0.3887  1.2330
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)    5.296      1.174    4.51  9.3e-05 ***
+ plaspro       -0.110      0.198   -0.56    0.58
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 0.628 on 30 degrees of freedom
+ Multiple R-squared:  0.0102, Adjusted R-squared: -0.0228
+ F-statistic: 0.31 on 1 and 30 DF, p-value: 0.582

```

```

> anova(m1.visc,mv.visc)

+ Analysis of Variance Table
+
+ Model 1: viscosity ~ pcv
+ Model 2: viscosity ~ pcv + plasfib + plaspro
+   Res.Df    RSS Df Sum of Sq    F Pr(>F)
+ 1       30 2.721
+ 2       28 2.583  2    0.1384 0.75  0.482

> anova(m2.visc,mv.visc)

+ Analysis of Variance Table
+
+ Model 1: viscosity ~ plasfib
+ Model 2: viscosity ~ pcv + plasfib + plaspro
+   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
+ 1       30 9.451
+ 2       28 2.582  2    6.869 37.24 1.29e-08 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(m3.visc,mv.visc)

+ Analysis of Variance Table
+
+ Model 1: viscosity ~ plaspro
+ Model 2: viscosity ~ pcv + plasfib + plaspro
+   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
+ 1       30 11.828
+ 2       28  2.582  2    9.246 50.12 5.59e-10 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

FACTOR PREDICTOR REGRESSION USING DUMMY VARIABLES

We can fit a factor predictor using **dummy variables**.

Suppose that a **factor predictor**, X , takes L levels, indexed by $l = 1, 2, \dots, L$. We proceed as follows:

1. Define L new "dummy" variables $X_1, \dots, X_L$, where, for $l = 1, \dots, L$,

$$X_l = \begin{cases} 1 & \text{if } X = l \\ 0 & \text{if } X \neq l \end{cases}$$

2. Fit the multiple regression model with $L - 1$ of the dummy variables as continuous co-variates, that is,

$$Y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_{L-1} x_{L-1} + \epsilon_i$$

Note that we cannot include all of $X_1, X_2, \dots, X_L$ if we have an intercept β_0 in the model; by default, R omits the first level of the factor, so we omit X_1 and regard 1 as the baseline group.

EXAMPLE Diabetes Data Set (more details in upcoming example).

Data set `Diabetes` has three subgroups defined by different patient characteristics, so $L = 3$.

A subset of the data are displayed below, with the new variables X_1 , X_2 and X_3 defined as above.

ID	glutest y	group x	Dummy 1 x_1	Dummy 2 x_2	Dummy 3 x_3
1	356	3	0	0	1
2	289	3	0	0	1
3	319	3	0	0	1
4	356	3	0	0	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
87	503	2	0	1	0
88	540	2	0	1	0
89	469	2	0	1	0
90	486	2	0	1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
113	1468	1	1	0	0
114	1487	1	1	0	0
115	714	1	1	0	0
116	1470	1	1	0	0

The three dummy variables will be used in the `lm` function.

Results: In the default parameterization:

```
> summary(lm(glutest~group,data=Diabetes))

+
+ Call:
+ lm(formula = glutest ~ group, data = Diabetes)
+
+ Residuals:
+    Min       1Q   Median       3Q      Max
+ -489.4  -44.0   -1.7   39.3  492.6
+
+ Coefficients:
+                Estimate Std. Error t value Pr(>|t|)
+ (Intercept)      1027.4       25.7    39.9  <2e-16 ***
+ groupChemically Diabetic  -533.4       35.4   -15.1  <2e-16 ***
+ groupNormal        -677.4       30.7   -22.1  <2e-16 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 146 on 141 degrees of freedom
+ Multiple R-squared:  0.777, Adjusted R-squared:  0.774
+ F-statistic: 246 on 2 and 141 DF,  p-value: <2e-16
```

Using the dummy variables:

```
> Y<-Diabetes$glutest
> X1<-as.numeric(Diabetes$group=="Overt Diabetic")
> X2<-as.numeric(Diabetes$group=="Chemically Diabetic")
> X3<-as.numeric(Diabetes$group=="Normal")
> summary(lm(Y~X2+X3))

+
+ Call:
+ lm(formula = Y ~ X2 + X3)
+
+ Residuals:
+    Min       1Q   Median       3Q      Max
+ -489.4  -44.0   -1.7   39.3  492.6
+
+ Coefficients:
+                Estimate Std. Error t value Pr(>|t|)
+ (Intercept)      1027.4       25.7    39.9  <2e-16 ***
+ X2              -533.4       35.4   -15.1  <2e-16 ***
+ X3              -677.4       30.7   -22.1  <2e-16 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 146 on 141 degrees of freedom
+ Multiple R-squared:  0.777, Adjusted R-squared:  0.774
+ F-statistic: 246 on 2 and 141 DF,  p-value: <2e-16
```

Replacing the intercept:

```
> summary(lm(glutest~-1+group,data=Diabetes))

+
+ Call:
+ lm(formula = glutest ~ -1 + group, data = Diabetes)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -489.4   -44.0    -1.7    39.3   492.6
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ groupOvert Diabetic    1027.4      25.7   39.9  <2e-16 ***
+ groupChemically Diabetic    493.9      24.3   20.4  <2e-16 ***
+ groupNormal              350.0      16.7   20.9  <2e-16 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 146 on 141 degrees of freedom
+ Multiple R-squared:  0.946, Adjusted R-squared:  0.944
+ F-statistic: 816 on 3 and 141 DF,  p-value: <2e-16
```

```
> summary(lm(Y~-1+X1+X2+X3))

+
+ Call:
+ lm(formula = Y ~ -1 + X1 + X2 + X3)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -489.4   -44.0    -1.7    39.3   492.6
+
+ Coefficients:
+      Estimate Std. Error t value Pr(>|t|)
+ X1    1027.4      25.7   39.9  <2e-16 ***
+ X2     493.9      24.3   20.4  <2e-16 ***
+ X3     350.0      16.7   20.9  <2e-16 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 146 on 141 degrees of freedom
+ Multiple R-squared:  0.946, Adjusted R-squared:  0.944
+ F-statistic: 816 on 3 and 141 DF,  p-value: <2e-16
```

FACTOR PREDICTOR REGRESSION

We need to take some care when combining factor predictors and covariates in the regression model. Suppose that we have only two predictors a **covariate**, x_1 and a **factor predictor**, x_2 , now taking L levels, with the levels being indexed by $l = 1, 2, \dots, L$.

EXAMPLE Binary Factor $L = 2$

Suppose that factor predictor x_2 takes two levels, labelled 0 and 1, that identify two data subgroups. Five models can be considered, that correspond to different straight-line models

- **MODEL 0 : Same intercept, slope zero**, in the two subgroups
- **MODEL 1 : Different intercept, slope zero**, in the two subgroups
- **MODEL 2 : Same intercept, same non-zero slope**, in the two subgroups
- **MODEL 3 : Different intercept, same non-zero slope**, in the two subgroups
- **MODEL 4 : Different intercept, different non-zero slopes**, in the two subgroups

The general model can be written

$$Y = \begin{cases} \beta_{00} + \beta_{01}x_1 + \epsilon & \text{GROUP 0 } (l = 0) \\ \beta_{10} + \beta_{11}x_1 + \epsilon & \text{GROUP 1 } (l = 1) \end{cases}$$

- **MODEL 0 :** $\beta_{00} = \beta_{10} = \beta_0, \beta_{01} = \beta_{11} = 0$
- **MODEL 1 :** $\beta_{00} \neq \beta_{10}, \beta_{01} = \beta_{11} = 0$
- **MODEL 2 :** $\beta_{00} = \beta_{10} = \beta_0, \beta_{01} = \beta_{11} = \beta_1 \neq 0$
- **MODEL 3 :** $\beta_{00} \neq \beta_{10}, \beta_{01} = \beta_{11} = \beta_1 \neq 0$
- **MODEL 4 :** $\beta_{00} \neq \beta_{10}, \beta_{01} \neq \beta_{11}$

The numbers of parameters, p , in each model are as follows:

MODEL 0	:	$p = 1$	β_0
MODEL 1	:	$p = 2$	β_{00}, β_{10}
MODEL 2	:	$p = 2$	$\beta_{00} = \beta_{10} = \beta_0, \beta_{01} = \beta_{11} = \beta_1$
MODEL 3	:	$p = 3$	$\beta_{00}, \beta_{10}, \beta_1$
MODEL 4	:	$p = 4$	$\beta_{00}, \beta_{10}, \beta_{01}, \beta_{11}$

The default parameterization used by R is different from the one described above. As in ANOVA, R takes a **baseline** group, and looks for **differences** in the parameters compared to the baseline group. The baseline group is taken to be the **first** listed subgroup for the factor predictor; in the binary example above, the baseline group would be Group 0. The interaction model is therefore

$$Y = [\beta_0 + (1 - x_2)\delta_{10}] + [(\beta_1 + (1 - x_2)\delta_{11})x_1] + \epsilon$$

- δ_{10} is the **change in intercept** from Group 0 to Group 1
- δ_{11} is the **change in slope** from Group 0 to Group 1

EXAMPLE Diabetes Data Set

The data in the data set `Diabetes` contain information on 68 diabetes patients falling into two clinically different categories (overt and chemical diabetics) and 76 normal controls. Measurements of plasma glucose in blood samples when fasting and in a dietary test are recorded.

The objective is to predict the test glucose levels from the fasting glucose levels in the three subgroups, and to find out if there is any significant difference between the subgroups.

In this analysis, there is a single response variable, one covariate and one factor predictor:

- y : **glutest**, the test glucose level
- x_1 : covariate **glufast**, the fasting glucose level
- x_2 : factor predictor **group**, the diabetes group
 - GROUP 1: Overt Diabetic
 - GROUP 2: Chemical Diabetic
 - GROUP 3: Normal Patients

We recode the baseline group to be 'Normal Patients'.

```
> Dia.copy<-Diabetes
> Dia.copy$group2<-Dia.copy$group
> levels(Dia.copy$group2)<-c('OD','CD','NO')
> Dia.copy$group2<-
+   factor(Dia.copy$group2, levels=c('NO','CD','OD'))
```

After recoding, the data set `Dia.copy` has levels

- GROUP 1: Normal Patients (NO)
- GROUP 2: Chemical Diabetic (CD)
- GROUP 3: Overt Diabetic (OD)

We decide to carry out the analysis on the **log** (ln) scale.

```
> attach(Diabetes)
> par(mar=c(4,4,1,0))
> plot(log(glufast), log(glutest), type="n")
> points(log(glufast[group=="Normal"]), log(glutest[group=="Normal"]),
+   pch=17)
> points(log(glufast[group=="Chemically Diabetic"]),
+   log(glutest[group=="Chemically Diabetic"]), pch=4)
> points(log(glufast[group=="Overt Diabetic"]),
+   log(glutest[group=="Overt Diabetic"]), pch=3)
> legend(4.2, 7.3, c("Normal", "Chem. Diabetic", "Overt Diabetic"),
+   pch=c(17, 4, 3))
```

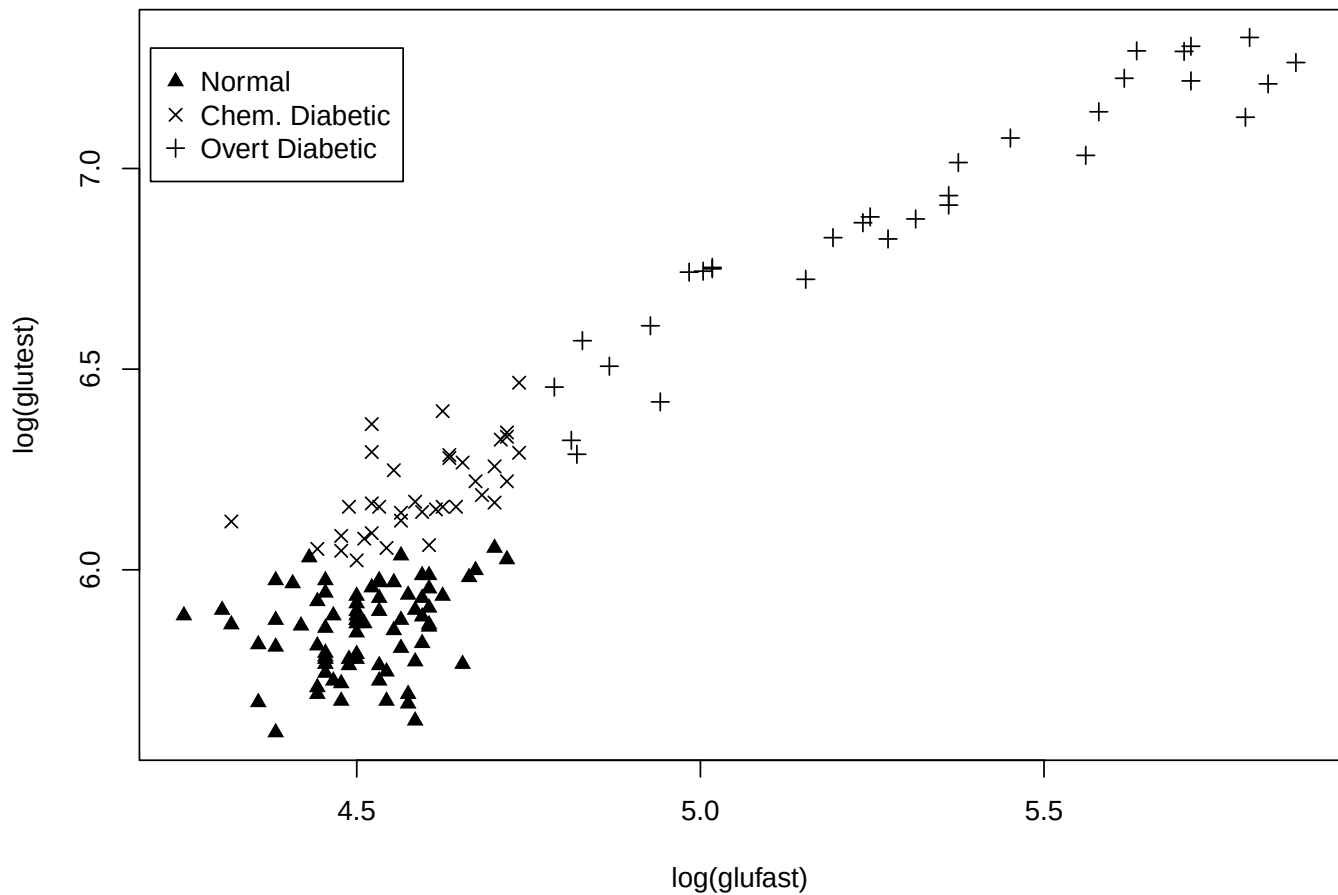


Figure 25: Diabetes data: raw data broken down by group

```
> detach(Diabetes)
```

```
> m.Diabetes<-lm(log(glutest)~log(glufast)*group2,data=Dia.copy)
>
> print(summary(aov(m.Diabetes)), digits = 5)
+               Df  Sum Sq Mean Sq    F value    Pr(>F)
+ log(glufast)      1 24.7664  24.7664 2593.8314 < 2.2e-16 ***
+ group2            2  2.2656   1.1328  118.6414 < 2.2e-16 ***
+ log(glufast):group2 2  0.1547   0.0773   8.0994 0.0004721 ***
+ Residuals       138  1.3177   0.0095
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> summary(m.Diabetes)
+
+ Call:
+ lm(formula = log(glutest) ~ log(glufast) * group2, data = Dia.copy)
+
+ Residuals:
+    Min       1Q   Median       3Q      Max
+ -0.2510 -0.0628  0.0067  0.0690  0.2155
+
```

```

+ Coefficients:
+               Estimate Std. Error t value Pr(>|t|)
+ (Intercept)      4.504      0.559   8.06  3.2e-13 ***
+ log(glufast)      0.299      0.124   2.41  0.0171 *
+ group2CD         -1.436      0.958  -1.50  0.1362
+ group2OD         -2.037      0.619  -3.29  0.0013 **
+ log(glufast):group2CD  0.382      0.210   1.82  0.0709 .
+ log(glufast):group2OD  0.535      0.134   4.00  0.0001 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 0.0977 on 138 degrees of freedom
+ Multiple R-squared:  0.954, Adjusted R-squared:  0.952
+ F-statistic: 569 on 5 and 138 DF,  p-value: <2e-16

```

The first ANOVA table demonstrates that there **is** a significant interaction between the covariate and the factor predictor ($F_{obs} = 8.099$, p-value 0.000472). This means that there is a **significantly different slope in at least two of the three** subgroups.

The second table gives the slope and intercept parameters in the three groups. The R parameterization is not directly in terms of the slopes and intercepts, but looks at **differences** from baseline subgroup, Group 1. For example, the Group 3 (Overt Diabetic) intercept and slope are

$$\text{INTERCEPT : } 4.504 + (-2.037) = 2.467 \quad \text{SLOPE : } 0.2991 + 0.5349 = 0.8340.$$

```

> attach(Diabetes)
> par(mfrow=c(2,2), mar=c(4,3,1,1))
> plot(log(glufast), log(glutest), type="n")
> points(log(glufast[group=="Normal"]), log(glutest[group=="Normal"]))
+ ,pch=17)
> be0<-coef(m.Diabetes)[1]; be1<-coef(m.Diabetes)[2]
> abline(be0,be1)
> title("Normal")
> legend(4.2,7.3,c("Normal"),pch=c(17))
>
> plot(log(glufast), log(glutest), type="n")
> points(log(glufast[group=="Chemically Diabetic"]),
+        log(glutest[group=="Chemically Diabetic"]),pch=4)
> be0<-coef(m.Diabetes)[1]+coef(m.Diabetes)[3];
> be1<-coef(m.Diabetes)[2]+coef(m.Diabetes)[5]
> abline(be0,be1)
> title("Chemically Diabetic")
> legend(4.2,7.3,c("Chem. Diabetic"),pch=c(4))
>
> plot(log(glufast), log(glutest), type="n")
> points(log(glufast[group=="Overt Diabetic"]),
+        log(glutest[group=="Overt Diabetic"]),pch=3)
> be0<-coef(m.Diabetes)[1]+coef(m.Diabetes)[4]
> be1<-coef(m.Diabetes)[2]+coef(m.Diabetes)[6]
> abline(be0,be1)
> title("Overt Diabetic")
> legend(4.2,7.3,c("Overt Diabetic"),pch=c(3))

```

```

>
> plot(log(glufast), log(glutest), type="n")
> be0<-coef(m.Diabetes)[1]; be1<-coef(m.Diabetes)[2]
> abline(be0,be1)
> be0<-coef(m.Diabetes)[1]+coef(m.Diabetes)[3]
> be1<-coef(m.Diabetes)[2]+coef(m.Diabetes)[5]
> abline(be0,be1,lty=2)
> be0<-coef(m.Diabetes)[1]+coef(m.Diabetes)[4]
> be1<-coef(m.Diabetes)[2]+coef(m.Diabetes)[6]
> abline(be0,be1,lty=3)
> legend(4.2,7.3,c("Normal","Chem. Diabetic","Overt Diabetic"),
+       lty=c(1,2,3))

```

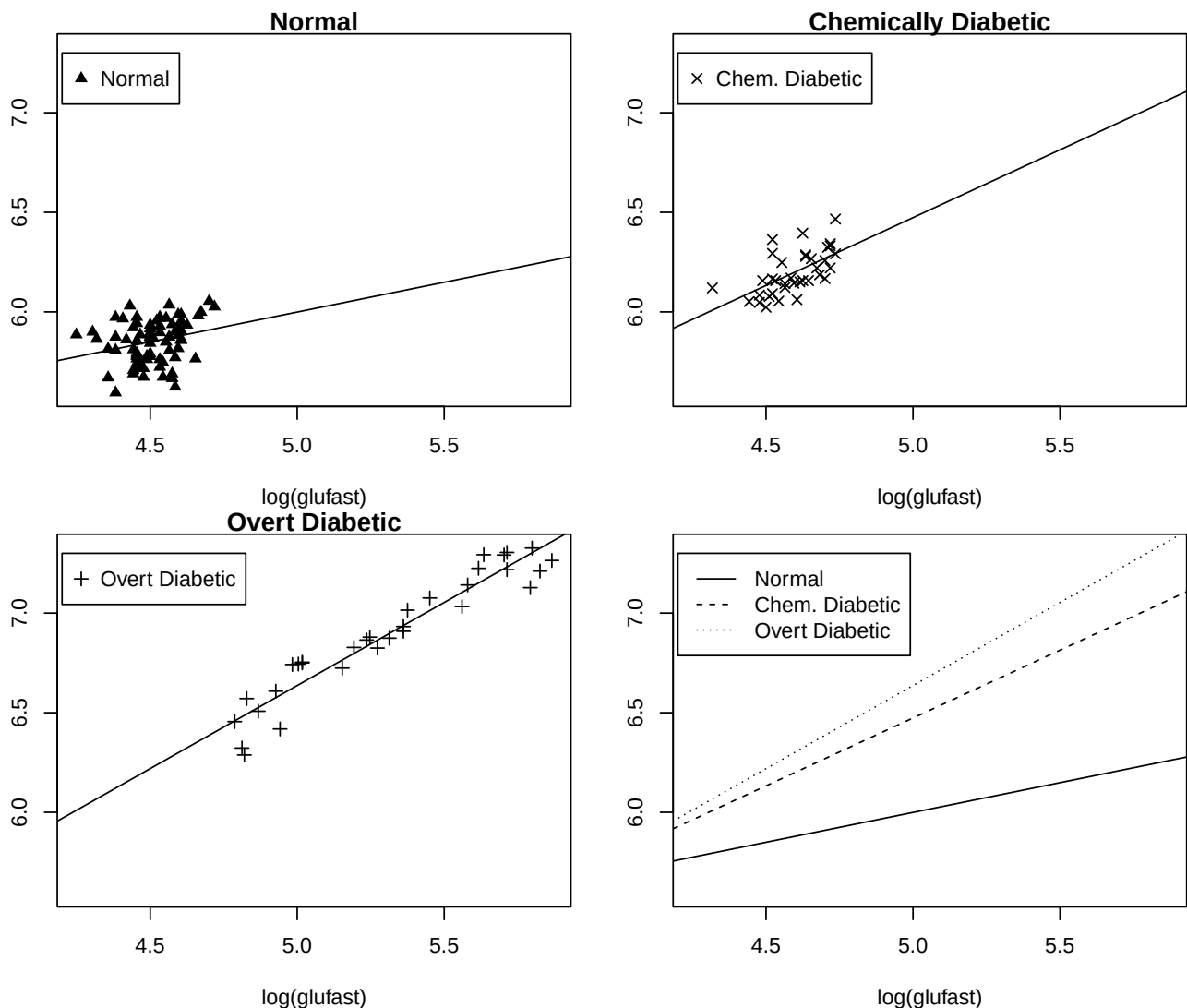


Figure 26: Diabetes data: fit broken down by group

```

> detach(Diabetes)

```

```

> drop1(m.Diabetes,test="F")

+ Single term deletions
+
+ Model:
+ log(glutest) ~ log(glufast) * group2
+
+               Df Sum of Sq   RSS   AIC F value    Pr(>F)
+ <none>                        1.318 -663.9
+ log(glufast):group2  2     0.1547 1.472 -651.9    8.099 0.000472 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The models that can be fitted are as follows:

```

MODEL 0 : Null model
MODEL 1 : Effect of group only
MODEL 2 : Effect of log(glufast) only
MODEL 3 : Additive main effect of group and log(glufast)
MODEL 4 : Main effects plus interaction

```

```

> m0.diabetes<-lm(log(glutest)~1,data=Dia.copy)
> m1.diabetes<-lm(log(glutest)~group2,data=Dia.copy)
> m2.diabetes<-lm(log(glutest)~log(glufast),data=Dia.copy)
> m3.diabetes<-lm(log(glutest)~group2+log(glufast),data=Dia.copy)
> m4.diabetes<-lm(log(glutest)~group2*log(glufast),data=Dia.copy)
>
> print(summary(aov(m0.diabetes)), digits = 5)

+               Df Sum Sq Mean Sq F value Pr(>F)
+ Residuals    143  28.504  0.19933

> print(summary(aov(m1.diabetes)), digits = 5)

+               Df Sum Sq Mean Sq F value    Pr(>F)
+ group2         2  24.344 12.1722  412.57 < 2.2e-16 ***
+ Residuals    141   4.160   0.0295
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> print(summary(aov(m2.diabetes)), digits = 5)

+               Df Sum Sq Mean Sq F value    Pr(>F)
+ log(glufast)   1  24.7664 24.7664  940.85 < 2.2e-16 ***
+ Residuals    142   3.7379   0.0263
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> print(summary(aov(m3.diabetes)), digits = 5)

```

```

+           Df  Sum Sq Mean Sq F value    Pr(>F)
+ group2      2 24.3443 12.1722 1157.43 < 2.2e-16 ***
+ log(glufast) 1  2.6877  2.6877  255.56 < 2.2e-16 ***
+ Residuals   140  1.4723  0.0105
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> print(summary(aov(m4.diabetes)), digits = 5)

+           Df  Sum Sq Mean Sq    F value    Pr(>F)
+ group2      2 24.3443 12.1722 1274.8150 < 2.2e-16 ***
+ log(glufast) 1  2.6877  2.6877  281.4841 < 2.2e-16 ***
+ group2:log(glufast) 2  0.1547  0.0773    8.0994 0.0004721 ***
+ Residuals   138  1.3177  0.0095
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

EXAMPLE Potato damage data

The potato damage data is an unbalanced three-way factorial design:

```
> head(Potato)
+      potato      regime  temp  photo  leak
+ 1 Variety 1 Room Temp  -4 C  4.6386  2.25
+ 2 Variety 1 Room Temp  -4 C  5.7914  4.34
+ 3 Variety 1 Room Temp  -4 C 29.3515  4.25
+ 4 Variety 1 Room Temp  -4 C 18.4173  6.14
+ 5 Variety 1 Room Temp  -4 C  2.2556  2.38
+ 6 Variety 1 Room Temp  -8 C  8.2358 16.30
> Potato.counts<-xtabs(~potato+regime+temp, data=Potato)
> ftable(Potato.counts)
+
+      temp  -4 C  -8 C
+ potato      regime
+ Variety 1 Room Temp      5      5
+      Cold Room      12      13
+ Variety 2 Room Temp      13      13
+      Cold Room      7      7
> levels(Potato$potato)<-c('V1','V2')
> levels(Potato$regime)<-c('Room','Cold')
> levels(Potato$temp)<-c('-4','-8')
```

```
> m.potato<-lm(leak~potato*regime*temp,data=Potato)
> summary(m.potato)
+
+ Call:
+ lm(formula = leak ~ potato * regime * temp, data = Potato)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -22.151  -3.325  -0.169   3.288  18.589
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)         3.87        3.85   1.01  0.31833
+ potatoV2            18.20        4.53   4.02  0.00015 ***
+ regimeCold          -1.53        4.58  -0.33  0.73913
+ temp-8              2.06        5.45   0.38  0.70673
+ potatoV2:regimeCold -18.12        6.11  -2.97  0.00417 **
+ potatoV2:temp-8      8.19        6.41   1.28  0.20567
+ regimeCold:temp-8    6.59        6.45   1.02  0.31065
+ potatoV2:regimeCold:temp-8 -9.44        8.61  -1.10  0.27670
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 8.61 on 67 degrees of freedom
+ Multiple R-squared:  0.64, Adjusted R-squared:  0.603
+ F-statistic: 17 on 7 and 67 DF, p-value: 1.02e-12
> print(summary(aov(m.potato)), digits = 5)
```

```

+               Df Sum Sq Mean Sq F value    Pr(>F)
+ potato               1 3420.2   3420.2  46.1178 3.578e-09 ***
+ regime               1 1951.8   1951.8  26.3173 2.674e-06 ***
+ temp                1 1255.6   1255.6  16.9302 0.0001084 ***
+ potato:regime        1 2089.9   2089.9  28.1805 1.349e-06 ***
+ potato:temp          1    28.9     28.9   0.3894 0.5347194
+ regime:temp          1     6.8      6.8   0.0916 0.7631284
+ potato:regime:temp   1    89.2     89.2   1.2027 0.2767010
+ Residuals          67 4968.9     74.2
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> drop1(m.potato, test="F")
+ Single term deletions
+
+ Model:
+ leak ~ potato * regime * temp
+               Df Sum of Sq  RSS    AIC F value Pr(>F)
+ <none>                                4969 330.5
+ potato:regime:temp 1         89.2 5058 329.8    1.203  0.277
> #Can drop 3-way interaction
>
> m1.potato<-update(m.potato, ~.-potato:regime:temp)
>
> drop1(m1.potato, test="F")
+ Single term deletions
+
+ Model:
+ leak ~ potato + regime + temp + potato:regime + potato:temp +
+       regime:temp
+               Df Sum of Sq  RSS    AIC F value    Pr(>F)
+ <none>                                5058 329.8
+ potato:regime 1         2093.2 7151 353.8   28.140 1.33e-06 ***
+ potato:temp   1           35.4 5094 328.4    0.477   0.492
+ regime:temp   1           6.8 5065 327.9    0.091   0.763
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> #Can drop 2-way interaction regime:temp
>
> m2.potato<-update(m1.potato, ~.-regime:temp)
> drop1(m2.potato, test="F")
+ Single term deletions
+
+ Model:
+ leak ~ potato + regime + temp + potato:regime + potato:temp
+               Df Sum of Sq  RSS    AIC F value    Pr(>F)
+ <none>                                5065 327.9
+ potato:regime 1         2094.6 7159 351.9   28.535 1.12e-06 ***
+ potato:temp   1           28.9 5094 326.4    0.393   0.533
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

> #Can drop 2-way interaction potato:temp
>
> m3.potato<-update(m2.potato,~.-potato:temp)
> drop1(m3.potato,test="F")
+ Single term deletions
+
+ Model:
+ leak ~ potato + regime + temp + potato:regime
+           Df Sum of Sq  RSS   AIC F value    Pr(>F)
+ <none>                    5094 326.4
+ temp           1       1226 6319 340.5    16.84 0.000108 ***
+ potato:regime  1       2090 7184 350.2    28.72 1.02e-06 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> summary(m3.potato)
+
+ Call:
+ lm(formula = leak ~ potato + regime + temp + potato:regime, data = Potato)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -21.070  -4.346  -0.008   3.772  19.670
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)      0.857      2.872    0.30  0.76619
+ potatoV2         22.296      3.174    7.02  1.1e-09 ***
+ regimeCold        1.771      3.192    0.55  0.58082
+ temp-8           8.087      1.971    4.10  0.00011 ***
+ potatoV2:regimeCold -22.854      4.264   -5.36  1.0e-06 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 8.53 on 70 degrees of freedom
+ Multiple R-squared:  0.631, Adjusted R-squared:  0.61
+ F-statistic: 29.9 on 4 and 70 DF, p-value: 1.59e-14
> print(summary(aov(m3.potato)), digits = 5)
+           Df Sum Sq Mean Sq F value    Pr(>F)
+ potato      1 3420.2  3420.2  47.002 2.286e-09 ***
+ regime      1 1951.8  1951.8  26.822 2.048e-06 ***
+ temp        1 1255.6  1255.6  17.255 9.096e-05 ***
+ potato:regime 1 2089.9  2089.9  28.721 1.015e-06 ***
+ Residuals   70 5093.7    72.8
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

EXAMPLE Task Distraction data

The task distraction data is a balanced two-way factorial design with a continuous covariate:

```
> head(TaskDistract)
+      Group          Task Distract Errors
+ 1 NonSmoker Pattern Recognition      107      9
+ 2 NonSmoker Pattern Recognition      133      8
+ 3 NonSmoker Pattern Recognition      123     12
+ 4 NonSmoker Pattern Recognition       94     10
+ 5 NonSmoker Pattern Recognition       83      7
+ 6 NonSmoker Pattern Recognition       86     10

> TD<-TaskDistract
> levels(TD$Group)<-c('Non','Del','Act')
> levels(TD$Task)<-c('Pat','Cog','Dri')
> TD.counts<-xtabs(~Group+Task, data=TD)
> ftable(TD.counts)
+      Task Pat Cog Dri
+ Group
+ Non      15  15  15
+ Del      15  15  15
+ Act      15  15  15

> m.Task<-lm(Errors~Group*Task*Distract,data=TD)
> print(summary(aov(m.Task)), digits = 5)
+              Df  Sum Sq Mean Sq  F value    Pr(>F)
+ Group          2    354.5   177.3    3.6645  0.028601 *
+ Task           2 28661.5 14330.8 296.2361 < 2.2e-16 ***
+ Distract       1  5747.0  5747.0 118.7986 < 2.2e-16 ***
+ Group:Task     4   1626.5   406.6    8.4055 5.374e-06 ***
+ Group:Distract 2    496.1   248.1    5.1277 0.007334 **
+ Task:Distract  2   2643.3  1321.6   27.3201 1.835e-10 ***
+ Group:Task:Distract 4    142.9    35.7    0.7386 0.567489
+ Residuals    117  5660.0    48.4
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> drop1(m.Task,test="F")
+ Single term deletions
+
+ Model:
+ Errors ~ Group * Task * Distract
+              Df Sum of Sq  RSS   AIC F value Pr(>F)
+ <none>                5660 540.3
+ Group:Task:Distract  4    142.9 5803 535.7    0.739  0.567
> #Can drop 3-way interaction
>
> m1.Task<-update(m.Task,~.-Group:Task:Distract)
> drop1(m1.Task,test="F")
+ Single term deletions
+
+ Model:
```

```

+ Errors ~ Group + Task + Distract + Group:Task + Group:Distract +
+   Task:Distract
+           Df Sum of Sq  RSS    AIC F value    Pr(>F)
+ <none>                        5803 535.7
+ Group:Task          4      1824.5 7627 564.6    9.511 1.01e-06 ***
+ Group:Distract      2        414.4 6217 541.0    4.320 0.0154 *
+ Task:Distract       2      2643.3 8446 582.4   27.558 1.37e-10 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> #Maybe drop Group:Distract
> m2.Task<-update(m1.Task,~.-Group:Distract)
> drop1(m2.Task,test="F")
+ Single term deletions
+
+ Model:
+ Errors ~ Group + Task + Distract + Group:Task + Task:Distract
+           Df Sum of Sq  RSS    AIC F value    Pr(>F)
+ <none>                        6217 541.0
+ Group:Task          4      1754 7971 566.6    8.675 3.35e-06 ***
+ Task:Distract       2      2725 8942 586.1   26.955 1.96e-10 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

> summary(m1.Task)
+
+ Call:
+ lm(formula = Errors ~ Group + Task + Distract + Group:Task +
+   Group:Distract + Task:Distract, data = TD)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -16.267  -3.019  -0.467   3.807  23.022
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)    7.67787    8.62606    0.89  0.37519
+ GroupDel      -20.44543    8.96349   -2.28  0.02430 *
+ GroupAct       0.66186    8.75109    0.08  0.93984
+ TaskCog      -33.54794    8.45784   -3.97  0.00012 ***
+ TaskDri       -2.54922    9.59524   -0.27  0.79094
+ Distract       0.01550    0.07593    0.20  0.83864
+ GroupDel:TaskCog 13.68795    3.59278    3.81  0.00022 ***
+ GroupAct:TaskCog 13.92341    3.64937    3.82  0.00022 ***
+ GroupDel:TaskDri -1.57046    3.65903   -0.43  0.66854
+ GroupAct:TaskDri -7.42416    3.72894   -1.99  0.04874 *
+ GroupDel:Distract 0.18410    0.07711    2.39  0.01852 *
+ GroupAct:Distract -0.00149    0.07466   -0.02  0.98408
+ TaskCog:Distract 0.45529    0.07133    6.38  3.3e-09 ***
+ TaskDri:Distract 0.02543    0.08105    0.31  0.75425
+ ---

```

```

+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 6.93 on 121 degrees of freedom
+ Multiple R-squared:  0.872, Adjusted R-squared:  0.858
+ F-statistic: 63.4 on 13 and 121 DF,  p-value: <2e-16
> print(summary(aov(m1.Task)), digits = 5)
+
+      Df  Sum Sq Mean Sq  F value    Pr(>F)
+ Group      2    354.5   177.3    3.6964 0.027656 *
+ Task       2 28661.5 14330.8 298.8182 < 2.2e-16 ***
+ Distract   1  5747.0  5747.0 119.8341 < 2.2e-16 ***
+ Group:Task  4   1626.5   406.6    8.4788 4.587e-06 ***
+ Group:Distract 2    496.1   248.1    5.1723 0.006991 **
+ Task:Distract 2   2643.3  1321.6   27.5582 1.373e-10 ***
+ Residuals 121   5802.9    48.0
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> m3.Task<-lm(Errors~Group+Task+Distract,data=TD)
> summary(m3.Task)
+
+ Call:
+ lm(formula = Errors ~ Group + Task + Distract, data = TD)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -25.67  -5.51  -1.07    5.09   27.02
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept) -28.8305     4.6692  -6.17    8e-09 ***
+ GroupDel      4.4677     1.9197   2.33    0.022 *
+ GroupAct      4.4007     1.9093   2.30    0.023 *
+ TaskCog     27.5102     1.9180  14.34 <2e-16 ***
+ TaskDri     -1.8414     1.9160  -0.96    0.338
+ Distract      0.3162     0.0378   8.38    8e-14 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 9.05 on 129 degrees of freedom
+ Multiple R-squared:  0.767, Adjusted R-squared:  0.758
+ F-statistic: 84.9 on 5 and 129 DF,  p-value: <2e-16

```

PREDICTION FROM GENERAL LINEAR MODELS

Consider the analysis presented includes in a summary of parameter estimates obtained for the Task Distraction data. The model being fitted is

$$\text{Group} + \text{Task} + \text{Distract} + \text{Group.Task} + \text{Task.Distract}$$

where

- `Group` is a factor predictor with three levels
- `Task` is a factor predictor with three levels
- `Distract` is a covariate

We know that `Group` and `Task` are factor predictors as the entries in the parameter estimates table appear in the form displaying a parameter estimate for each level. etc. R takes the first level for each factor to be the baseline level; in this case, each factor has three levels, so the baseline treatment group is the one where

$$\text{Group} = \text{'NonSmoker'} \quad \text{and} \quad \text{Task} = \text{'Pattern Recognition'}$$

To make a prediction from this model, we need to combine the parameter estimates. We do this by considering the model formula, and extracting one numerical value for each term, remembering to also add in the intercept.

Suppose we want to make predictions for

- **Prediction A:** Subject in Group "NonSmoker", doing Task "Driving Simulation", with Distraction level 180.
- **Prediction B:** Subject in Group "Delayed Smoker", doing Task "Cognitive Task", with Distraction level 180.
- **Prediction C:** Subject in Group "Active Smoker", doing Task "Pattern Recognition", with Distraction level 100.

We can go through term by term in the model, as in the following table :

Term	Prediction A	Prediction B	Prediction C
Intercept	7.678	7.678	7.678
Group	0.000	-20.445	0.662
Task	-2.549	-33.548	0.000
Distract	0.015×180	0.015×180	0.015×100
Group.Task	0.000	13.688	0.000
Task.Distract	0.025×180	0.455×180	0.000×100
Total	12.495	85.251	9.740

This can be achieved in R using the `predict` function, and a `newdata` data frame that specifies the values at which the predictions are made. We may compute confidence intervals for the predicted value excluding or including the residual variation.

```

> newTask<-data.frame(Group=c("Non", "Del", "Act"),
+   Task=c("Dri", "Cog", "Pat"), Distract=c(180, 180, 100))
> predict(m1.Task, newdata=newTask, interval="confidence")

+   fit      lwr      upr
+ 1 12.49529  2.71407 22.2765
+ 2 85.25114 76.35698 94.1453
+ 3  9.74008  5.74588 13.7343

> predict(m1.Task, newdata=newTask, interval="prediction")

+   fit      lwr      upr
+ 1 12.49529 -4.34640 29.3370
+ 2 85.25114 68.90868 101.5936
+ 3  9.74008 -4.54011 24.0203

```

Notes:

- when you include covariate terms, remember to multiply the coefficient by the value at which you are predicting; here we multiply the coefficients involving Distract by 180 (in Predictions A and B) and 100 (in Prediction C).
- this approach of going through term by term works for all models. For example, if we fitted the model

Group + Task + Distract

which is model m3.Task, and obtained the parameter estimates

```

> summary(m3.Task)
+
+ Call:
+ lm(formula = Errors ~ Group + Task + Distract, data = TD)
+
+ Residuals:
+   Min       1Q   Median       3Q      Max
+ -25.67  -5.51  -1.07   5.09  27.02
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept) -28.8305    4.6692   -6.17   8e-09 ***
+ GroupDel      4.4677    1.9197    2.33   0.022 *
+ GroupAct      4.4007    1.9093    2.30   0.023 *
+ TaskCog      27.5102    1.9180   14.34  <2e-16 ***
+ TaskDri     -1.8414    1.9160   -0.96   0.338
+ Distract      0.3162    0.0378    8.38   8e-14 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 9.05 on 129 degrees of freedom
+ Multiple R-squared:  0.767, Adjusted R-squared:  0.758
+ F-statistic: 84.9 on 5 and 129 DF,  p-value: <2e-16

```

then for a prediction in Group 'Delayed Smoker', Task 'Driving Simulation' and Distraction 140, we would have

$$-28.830 + 4.466 + (-1.841) + (0.316 \times 140) = 18.035$$

For predictions A, B and C, we have

```
> predict(m3.Task, newdata=newTask, interval="confidence")
+      fit      lwr      upr
+ 1 26.2425 19.99076 32.4942
+ 2 60.0619 54.06667 66.0570
+ 3  7.1893  3.60742 10.7712
> predict(m3.Task, newdata=newTask, interval="prediction")
+      fit      lwr      upr
+ 1 26.2425  7.27409 45.2109
+ 2 60.0619 41.17647 78.9472
+ 3  7.1893 -11.07393 25.4525
```

Note that predictions from the full factorial model

Group*Task*Distract

(fitted as m.Task) do not differ greatly from the predictions in the selected model.

```
> predict(m.Task, newdata=newTask, interval="confidence")
+      fit      lwr      upr
+ 1 12.88873 -0.195647 25.9731
+ 2 88.54081 77.868781 99.2128
+ 3  8.46897  4.122008 12.8159
> predict(m.Task, newdata=newTask, interval="prediction")
+      fit      lwr      upr
+ 1 12.88873 -6.10970 31.8872
+ 2 88.54081 71.11578 105.9658
+ 3  8.46897 -5.97525 22.9132
```

- Making predictions near and beyond the end of the Distraction range might lead to inappropriate (for example, negative-valued) predictions. Here the minimum value of Distraction is 64, the maximum is 191.

SUMMARY OF ISSUES IN ANOVA, REGRESSION AND LINEAR MODELLING

1. **Model Assumptions :** The key model assumption is that the residual (measurement) errors are independent and identically distributed Normal random quantities. If this assumption is not met, then none of the hypothesis tests based on the t- and F-distributions are valid.

The validity of this model assumption can be checked by the inspection of the *residuals*, $\hat{e}_i$, or *standardized residuals*, $\hat{z}_i$ where

$$\hat{e}_i = y_i - \hat{y}_i \qquad \hat{z}_i = \frac{\hat{e}_i}{s} = \frac{y_i - \hat{y}_i}{s}$$

for $i = 1, \dots, n$. Plots of the residuals can be used to check for

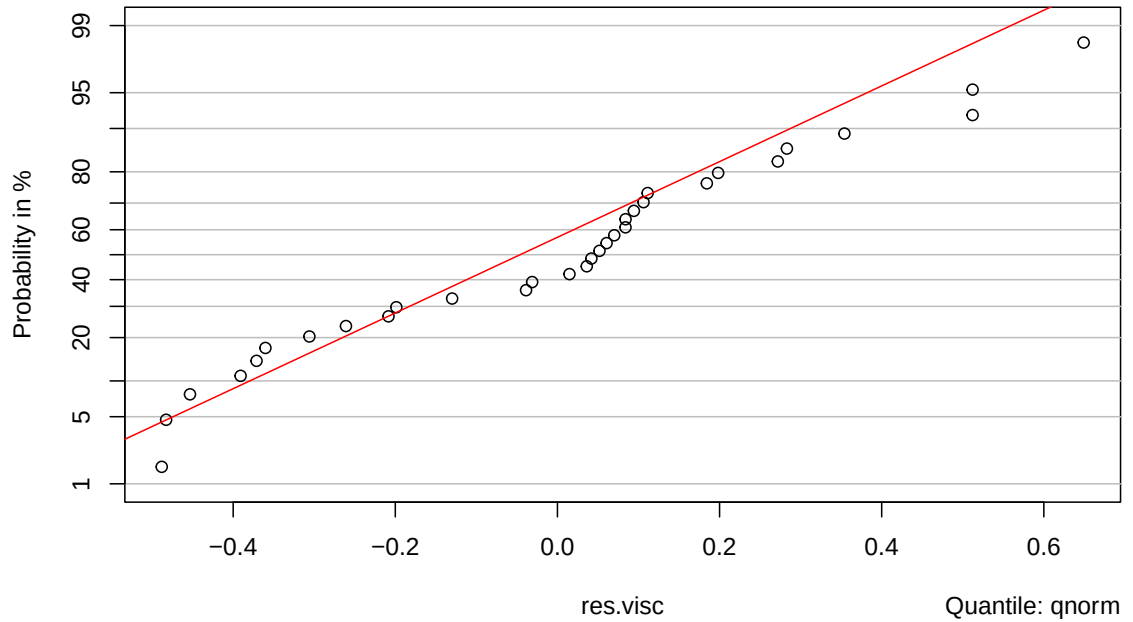
- (i) *Normality* To check the normality of the residuals, a *histogram* or *probability plot* can be used. A probability plot is a plot constructed using the **observed** (standardized) residuals and their **theoretical** counterparts **assuming a normal model**. Points in such a plot should lie on a straight-line with slope one; any deviation from this may indicate deviation from normality.

The difference between P-P plots and Q-Q plots is that the P-P plots use the observed cumulative probability (that is, plots the observed residuals against the **rank** of the observed residuals divided by $n + 1$), whereas Q-Q plots plot theoretical quantiles against observed quantiles from the residual distribution(that is, the $i/(n + 1)$ quantiles for $i = 1, \dots, n$ against the observed residuals sorted into ascending order).

The examples below are from the Viscosity vs PCV data, and the straight-line and quadratic model analysis of the Hooker data. For the Viscosity data, the probability plot indicates that the residual variance is larger in the middle compared to the of the residual range. For the Hooker data straight-line fit, the points in the probability plot do not lie on the straight-line, so again a deviation from normality is indicated; for the quadratic fit, the plot indicates normality of the residuals.

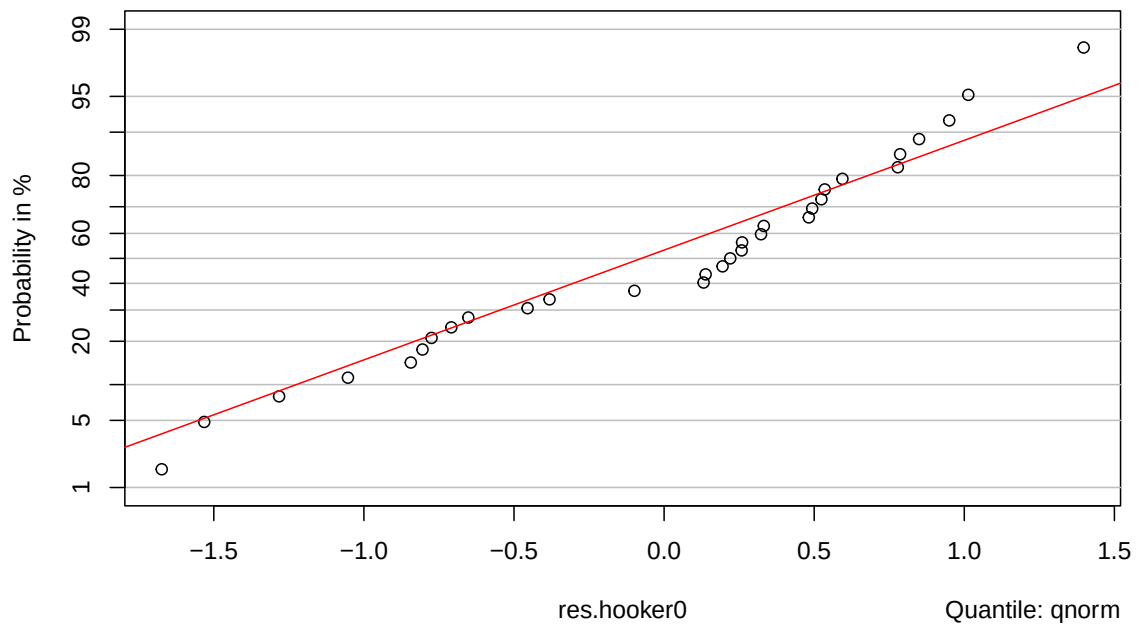
```
> m0.visc<-lm(viscosity~pcv,data=Viscosity)
> m0.hooker<-lm(temp~pressure,data=Hooker)
> m1.hooker<-lm(temp~pressure+I(pressure^2),data=Hooker)
>
> res.visc<-residuals(m0.visc)
> res.hooker0<-residuals(m0.hooker)
> res.hooker1<-residuals(m1.hooker)
>
> #P-P plots
> #library(e1071)
> e1071::probplot(res.visc)
> title('P-P plot for Viscosity residuals')
```

P-P plot for Viscosity residuals



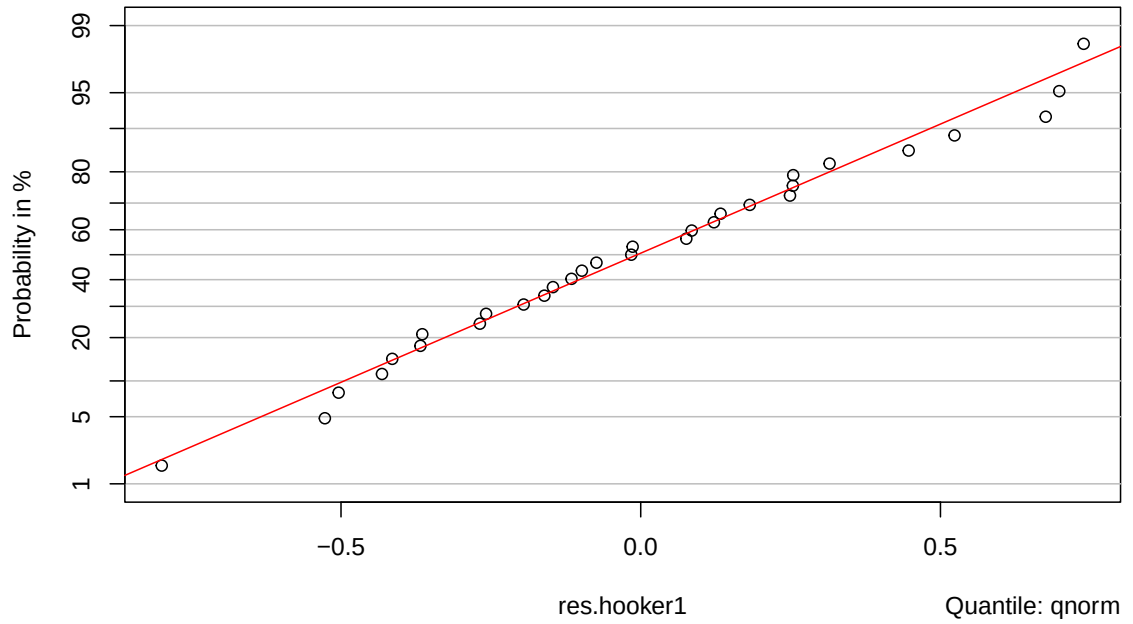
```
> e1071::probplot(res.hooker0)
> title('P-P plot for Linear fit to Hooker data')
```

P-P plot for Linear fit to Hooker data



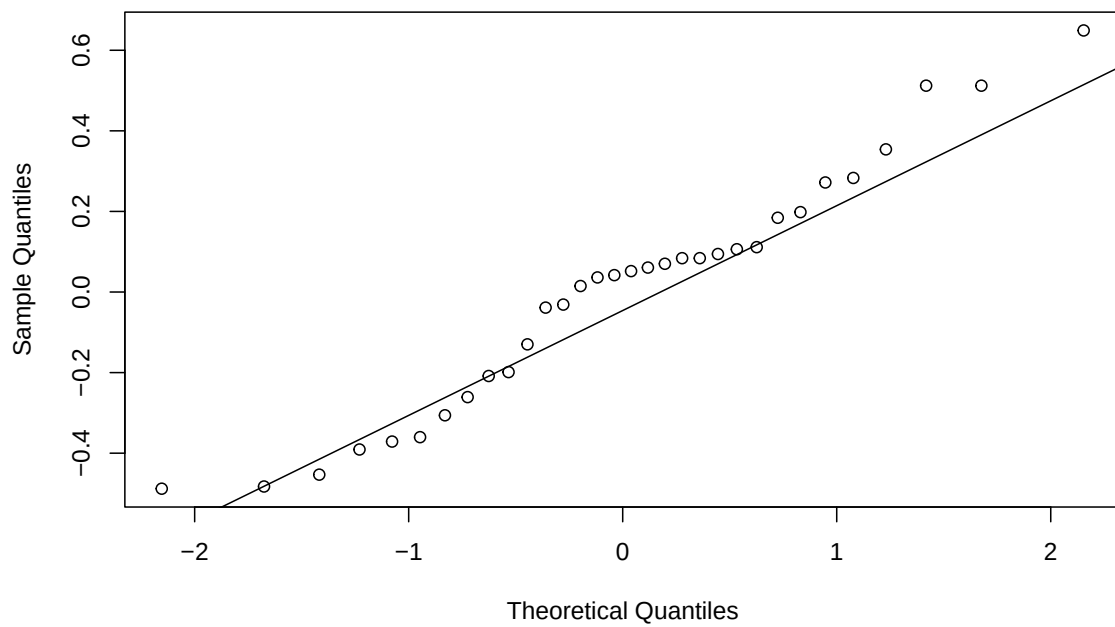
```
> e1071::probplot(res.hooker1)
> title('P-P plot for Quadratic fit to Hooker data')
```

P-P plot for Quadratic fit to Hooker data

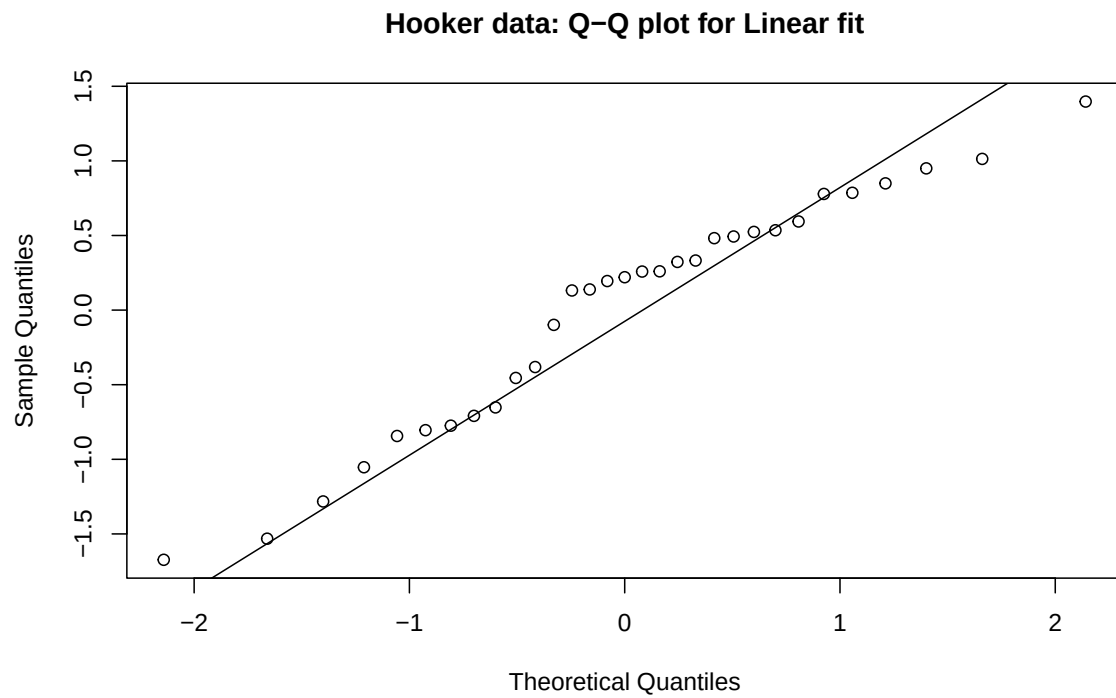


```
> #Q-Q plots  
> qqnorm(res.visc, main='Viscosity data: Q-Q plot for residuals')  
> qqline(res.visc)
```

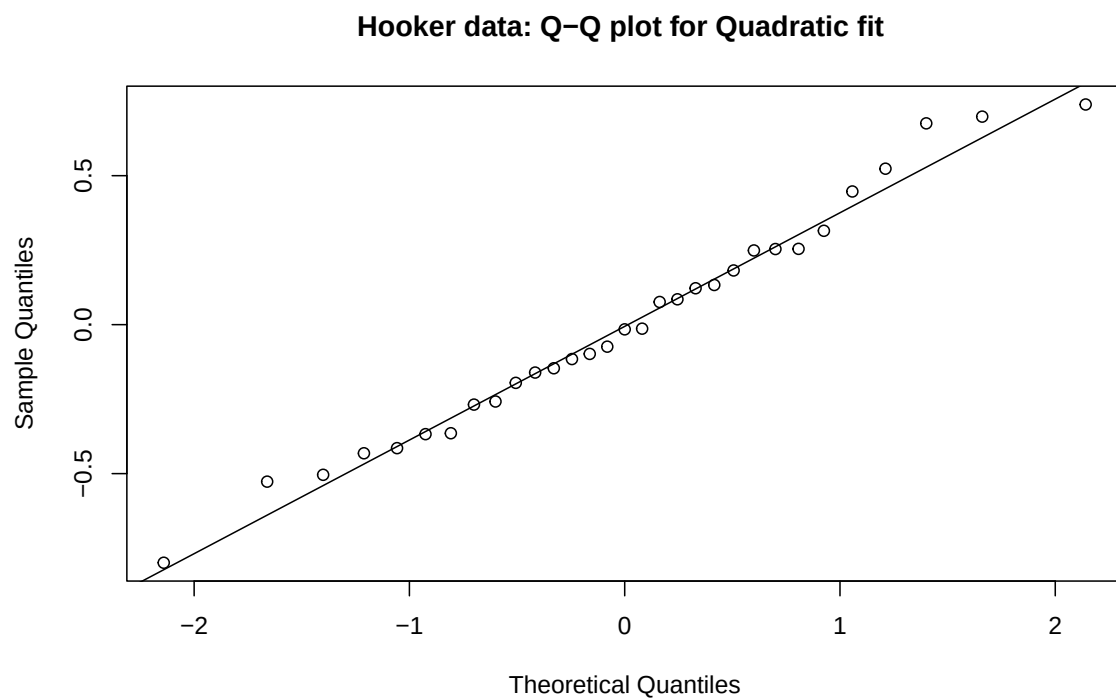
Viscosity data: Q-Q plot for residuals



```
> qqnorm(res.hooker0, main='Hooker data: Q-Q plot for Linear fit')
> qqline(res.hooker0)
```



```
> qqnorm(res.hooker1, main='Hooker data: Q-Q plot for Quadratic fit ')
> qqline(res.hooker1)
```



(ii) *Dependence on the covariates*

(iii) *Constant variance*

(iv) *Outliers*: An *outlier* is an response value that gives rise to a residual which is large in magnitude, indicating that the fit of the model is poor for that data point.

Outliers can significantly alter the fit of a model, and the parameter estimates. If an outlier is suspected, then careful consideration should be given to omitting that data point from the analysis (see below; estimates (standard errors) change in the presence of an outlier).

2. **Data Transformations** : The response variable and continuous covariates can be **transformed** (using log or square-root transformation say) to improve the fit of the model, or to make the model assumptions more appropriate.
3. **Model Selection** : Model selection by means of stepwise selection and sequential ANOVA-F testing can be an effective way of finding the important explanatory variables and interactions. However it must be carried out with care. In general, we aim to select the **simplest model** that provides an adequate fit to the data. The goodness of fit measures R^2 and adjusted R^2 statistics can provide a final assessment of model adequacy.
4. **Multicollinearity** : *Multicollinearity* is the term to describe dependence between the covariates used in a regression model. If the covariates are highly correlated, then the estimated coefficients for those covariates in a multiple regression need careful interpretation.

If two covariates are highly correlated, then if one is a useful predictor of the response, the other will likely appear to be a useful predictor as well, that is if one estimated coefficient is significantly different from zero, then the other will be also. However, in a multiple regression model with both covariates included, it might be that neither coefficient is significantly different from zero.

5. **Predicting outside the Range of the Covariates** : In a regression model, the fitted parameters reflect relationships and dependencies in the *observed* data. The model can be used for prediction, but is only likely to be reliable if the prediction is carried out at x values within the range of the observed x s.

For example, in a simple linear regression, if x takes values on the range $(0, 100)$, predictions at new x values within this range will be reliable, but predictions at, say, $x = 200$ will be much less reliable.