Problem 1

We run the supplied code that reads the data into either SAS or R. Then to get a 1-way MANOVA we run the following:

```
* PROBLEM 1;
proc glm data=breakfast;
class type mfr;
model calories protein fat sodium fiber carbo sugars potass rating = mfr
/ i ss3;
manova h=_all_/printe printh;
run;
```

R code 1:

```
dep_vars <-
cbind(breakfast$calories,breakfast$protein,breakfast$fat,breakfast$sodium,breakf
ast$fiber,breakfast$carbo,breakfast$sugars,breakfast$potass,breakfast$rating)
lmfit <- manova(dep_vars ~ mfr,data=breakfast)
Ano.fit <- car::Anova(lmfit,test = 'Wilks')
Ano.fit
Ano.fit$SSPE
Ano.fit$SSP
```

R code 2:

```
manova <- manova(cbind(calories, protein, fat, sodium, fiber, carbo, sugars, potass, rating) ~ mfr,
data=breakfast)
manova_summary  <- summary(manova)

################### wilks_lambda ####################################
# Extract and print values with names
wilks_lambda_test <-summary(manova, test = "Wilks")
wilks_lambda <- wilks_lambda_test$stats['mfr', "Wilks"]
approx_f_Wilks_Lambda <- wilks_lambda_test$stats['mfr', 'approx F']
num_df_Wilks_Lambda <- wilks_lambda_test$stats['mfr', 'num Df']
den_df_Wilks_Lambda <- wilks_lambda_test$stats['mfr', 'den Df']
p_value_Wilks_Lambda <- wilks_lambda_test$stats['mfr', 'Pr(>F)']
```

Question 1.1

We find in the output

SAS

MANOVA Test Criteria and F Approximations for the Hypothesis of No Overall mfr Effect
H = Type III SSCP Matrix for mfr
E = Error SSCP Matrix

S=5 M=1.5 N=30

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.16609614	3.08	45	280.44	<.0001
Pillai's Trace	1.29939625	2.57	45	330	<.0001
Hotelling-Lawley Trace	2.65598925	3.58	45	179.59	<.0001
Roy's Greatest Root	1.56581966	11.48	9	66	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.

R code 1:

Type II MANOVA Tests: Wilks test statistic
Df test stat approx F num Df den Df Pr(>F)
mfr 5 0.1661 3.0774 45 280.44 7.46e-09 ***

R code 2:

Wilks' Lambda: 0.1660961
Wilks' Lambda Approximate F-value: 3.077347
Wilks' Lambda Num DF: 45
Wilks' Lambda Denominator DF: 280.4439
Wilks' Lambda p-value: 7.459968e-09

0.1661

Question 1.2

We consider the output

SAS

H = Type III SSCP Matrix for mfr

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	rating
calories	4661.7646719	-65.07824238	63.452314911	28092.599623	-215.8093463	473.84081085	886.18276587	-1820.524178	-3931.198217
protein	-65.07824238	2.0266942884	-3.711537573	-419.1872385	8.0841322262	0.2784249821	-14.75951159	86.322761367	89.180717143
fat	63.452314911	-3.711537573	15.695449923	486.45545847	-27.65339667	-30.90690082	14.38127846	-434.13745	-203.9313674
sodium	28092.599623	-419.1872385	486.45545847	180642.56589	-1667.901128	2846.3098808	5298.6747769	-17952.73999	-25746.87548
fiber	-215.8093463	8.0841322262	-27.65339667	-1667.901128	54.812670209	35.780004276	-44.70906086	833.42564692	453.12189079
carbo	473.84081085	0.2784249821	-30.90690082	2846.3098808	35.780004276	278.12342822	-12.69317026	673.74194462	181.6550628
sugars	886.18276587	-14.75951159	14.38127846	5298.6747769	-44.70906086	-12.69317026	246.57676016	-262.9222893	-949.6448722
potass	-1820.524178	86.322761367	-434.13745	-17952.73999	833.42564692	673.74194462	-262.9222893	14329.937883	5769.7310884
rating	-3931.198217	89.180717143	-203.9313674	-25746.87548	453.12189079	181.6550628	-949.6448722	5769.7310884	5373.6966242

R code 1:

```
        [,1]      [,2]      [,3]      [,4]       [,5]      [,6]      [,7]       [,8]       [,9]
[1,] 4661.76467 -65.078242  63.452315 28092.5996 -215.809346 473.840811 886.18277 -1820.52418 -3931.19822
[2,]  -65.07824   2.026694  -3.711538  -419.1872    8.084132   0.278425 -14.75951    86.32276    89.18072
[3,]   63.45231  -3.711538  15.695450   486.4555  -27.653397 -30.906901  14.38128  -434.13745  -203.93137
[4,] 28092.59962 -419.187239 486.455458 180642.5659 -1667.901128 2846.309881 5298.67478 -17952.73999 -25746.87548
[5,] -215.80935   8.084132 -27.653397 -1667.9011   54.812670  35.780004 -44.70906   833.42565   453.12189
[6,]  473.84081   0.278425 -30.906901  2846.3099   35.780004 278.123428 -12.69317   673.74194   181.65506
[7,]  886.18277 -14.759512  14.381278  5298.6748  -44.709061 -12.693170 246.57676  -262.92229  -949.64487
[8,] -1820.52418  86.322761 -434.137450 -17952.7400  833.425647 673.741945 -262.92229 14329.93788 5769.73109
[9,] -3931.19822  89.180717 -203.931367 -25746.8755  453.121891 181.655063 -949.64487 5769.73109 5373.69662
```

R code 2 and output:

```
print("Type III SS matrix for manufacturers:")
wilks_lambda_test$SS$mfr
```

```
Type III SS matrix for manufacturers:
            calories     protein         fat      sodium       fiber       carbo      sugars       potass       rating
calories  4661.76467 -65.078242   63.452315  28092.5996 -215.809346  473.840811  886.18277 -1820.52418 -3931.19822
protein    -65.07824   2.026694   -3.711538   -419.1872    8.084132    0.278425  -14.75951    86.32276    89.18072
fat         63.45231  -3.711538   15.695450    486.4555  -27.653397  -30.906901   14.38128  -434.13745  -203.93137
sodium   28092.59962 -419.187239 486.455458 180642.5659 -1667.901128 2846.309881 5298.67478 -17952.73999 -25746.87548
fiber      -215.80935   8.084132  -27.653397  -1667.9011   54.812670   35.780004  -44.70906   833.42565   453.12189
carbo       473.84081   0.278425  -30.906901   2846.3099   35.780004  278.123428  -12.69317   673.74194   181.65506
sugars      886.18277 -14.759512   14.381278   5298.6748  -44.709061  -12.693170  246.57676  -262.92229  -949.64487
potass    -1820.52418  86.322761 -434.137450 -17952.7400  833.425647  673.741945 -262.92229 14329.93788  5769.73109
rating    -3931.19822  89.180717 -203.931367 -25746.8755  453.121891  181.655063 -949.64487  5769.73109  5373.69662
```

Question 1.3

We use

Theorem 4.25

The ratio test for the test of the hypothesis H_0 against H_1 is given by the critical region

$$\{y_{11}, \ldots, y_{kn_k} \mid \frac{\det(\mathbf{w})}{\det(\mathbf{t})} \leq U(p, k-1, n-k)_\alpha\}.$$

R code:

```
Q13_variables <- c("calories", "protein", "fat", "sodium", "fiber", "carbo", "sugars", "potass","rating")
Q13_num_variables  <-  length(Q13_variables)
cat("Number of manufacturers:", length(unique(breakfast$mfr)),"\n")
cat("Number of observations:", nrow(breakfast), "\n")
cat("Number of variables:", Q13_num_variables, "\n")
```

We have 9 variable: p = 9
We have 6 manufacturers: k = 6
We have 76 observations: n = 76

U(9,5,70)

Question 1.4

For this on we can either work directly in the U-distribution, or we can convert it to an F-distribution.

We use from page 296

Theorem 4.22

Let U be U(s,r,n-k)-distributed and let

$$t = \begin{cases} 1 & s^2+r^2=5 \\ \sqrt{\frac{s^2r^2-4}{s^2+r^2-5}} & s^2+r^2 \neq 5 \end{cases}$$
$$v = \frac{1}{2}(2(n-k)+r-s-1).$$

Then

$$\mathrm{F} = \frac{1-U^{\frac{1}{t}}}{U^{\frac{1}{t}}} \cdot \frac{vt+1-\frac{1}{2}sr}{sr}$$

is approximately distributed as

$$\mathrm{F}(sr, vt+1-\frac{1}{2}sr).$$

If either s or r are equal to 1 or 2, then the approximation is exact.

Since we have *U(s, r, n-k)=U(1,2,n-2)*, the approximation is exact.

Now it is just to plug in the numbers, and see when we have a P-value less than 0.05. This can be done in either SAS or R.

This is the case for n=6 where we have

t=1, v=4 and the f-value=8

We test in F(2,4) and get a p-value of 0.04

R code:

```
######## code for n=4,5,6,7,8,9 and print the results #########
# Define the observed test-statistic u
u <- 0.2

# Define the values of s, r, k, and a range of n values
s <- 1
r <- 2
k <- 2
n_values <- 4:9  # A range of n values
# Create empty vectors to store results
F_statistics <- numeric(length(n_values))
p_values <- numeric(length(n_values))
```

```
# Loop through different values of n
for (i in 1:length(n_values)) {
  n <- n_values[i]

  # Calculate v
  v <- 0.5 * (2 * (n - k) + r - s - 1)

  # Calculate t based on the given conditions
  t <- ifelse(s^2 + r^2 == 5, 1, sqrt((s^2 * r^2 - 4) / (s^2 + r^2 - 5)))

  # Calculate the F-statistic
  F_statistic <- (1 - u^(1/t)) / (u^(1/t)) * ((v * t + 1 - 0.5 * s * r)/(s * r))

  # Calculate the degrees of freedom for the F-distribution
  df1 <- s * r
  df2 <- v * t + 1 - 0.5 * s * r

  # Calculate the p-value using the CDF of the F-distribution
  p_value <- 1 - pf(F_statistic, df1, df2)

  # Store the results in vectors
  F_statistics[i] <- F_statistic
  p_values[i] <- p_value
}

# Create a data frame to display the results
results <- data.frame(n = n_values, F_statistic = F_statistics, p_value = p_values)

# Print the results
print(results)
```

R code output:

```
  n F_statistic     p_value
1 4           4 0.200000000
2 5           6 0.089442719
3 6           8 0.040000000
4 7          10 0.017888544
5 8          12 0.008000000
6 9          14 0.003577709
```

Problem 2

We run the following code

```
proc discrim data=breakfast pool=yes;
class mfr;
var protein fat sodium fiber carbo sugars potass calories;
```

```
run;
```

Question 2.1

We find in the output

Number of Observations and Percent Classified into mfr							
From mfr	G	K	N	P	Q	R	Total
G	14 63.64	0 0.00	0 0.00	2 9.09	1 4.55	5 22.73	22 100.00
K	2 8.70	9 39.13	3 13.04	3 13.04	1 4.35	5 21.74	23 100.00
N	0 0.00	0 0.00	5 83.33	1 16.67	0 0.00	0 0.00	6 100.00
P	0 0.00	2 22.22	0 0.00	6 66.67	0 0.00	1 11.11	9 100.00
Q	3 37.50	0 0.00	0 0.00	0 0.00	5 62.50	0 0.00	8 100.00
R	1 12.50	0 0.00	0 0.00	0 0.00	0 0.00	7 87.50	8 100.00
Total	20 26.32	11 14.47	8 10.53	12 15.79	7 9.21	18 23.68	76 100.00
Priors	0.16667	0.16667	0.16667	0.16667	0.16667	0.16667	

Column wise it is 2+3+1+2+3+2+3+1+1+1+5+5+1 = 30

R code:

```
library(MASS)


# Prior probabilities for the classes
# SAS by default has equal prior probabilities, so we need equal prior in R to receive the same results
different_manufacturers <- unique(breakfast$mfr)
num_manufacturers  <- length(different_manufacturers)
prior  <-  rep(1/num_manufacturers, num_manufacturers)

#linear discriminant analysis
# Define the classes (mfr)
breakfast$class <- as.factor(breakfast$mfr)
# Define the variables for the analysis
variables <- c("calories", "protein", "fat", "sodium", "fiber", "carbo", "sugars", "potass")
# Perform Linear Discriminant Analysis
z <- lda(class ~ ., data = breakfast[, c("class", variables)],prior=prior)


Class_Level_Information = data.frame("Frequency" = z$counts,"Proportion"=z$counts/z$N,"Prior"=z$prior)
print("Class Level Information:")
Class_Level_Information


n <- nrow(breakfast)
Classes <- nlevels(breakfast$mfr)
```

```
paste0("DF Within Classes = ",n-Classes)
paste0("DF Between Classes = ",Classes-1)

zpred <- predict(z)

#Confusion Matrix:
print("Confusion Matrix:")
xtabs(~breakfast$mfr+zpred$class)
```

R output:

```
                zpred$class
breakfast$mfr  G  K  N  P  Q  R
            G 14  0  0  2  1  5
            K  2  9  3  3  1  5
            N  0  0  5  1  0  0
            P  0  2  0  6  0  1
            Q  3  0  0  0  5  0
            R  1  0  0  0  0  7
> |
```

30

Question 2.2

We find in the output

Generalized Squared Distance to mfr						
From mfr	G	K	N	P	Q	R
G	0	4.22917	17.05810	3.12210	5.37016	1.98011
K	4.22917	0	8.16937	0.61728	11.64772	1.90281
N	17.05810	8.16937	0	8.50438	17.86481	10.80401
P	3.12210	0.61728	8.50438	0	8.43268	2.22019
Q	5.37016	11.64772	17.86481	8.43268	0	9.48675
R	1.98011	1.90281	10.80401	2.22019	9.48675	0

R code:

```
library(Rfast)
pcov  <- pooled.cov(as.matrix(breakfast[,variables]),breakfast$mfr)
Means <- as.matrix(z$means)
invCov <- solve(pcov)
# Extract unique levels from breakfast$mfr
unique_levels <- levels(breakfast$mfr)
num_col <- length(unique(breakfast$mfr))

# Create an empty matrix to store the Mahalanobis distances with the equal priors ################
maha <- matrix(c(rep(0, num_col^2)), ncol = num_col)
```

```
# Define the names for rows and columns (assuming unique_levels contains the names)
rownames(maha) <- unique_levels
colnames(maha) <- unique_levels

for (i in 1:num_col) {
  for (j in 1:num_col) {
    mu <- Means[i, ] - Means[j, ]
    maha[i, j] <- mu %*% invCov %*% mu
  }
}

# squared Mahalanobis distances between the 6 mfr types.
# maha    : Assuming equal priors

print("Generalized Squared Distance to species (equal priors):")
maha
```

R output:

	G	K	N	P	Q	R
G	0.000000	4.2291667	17.058099	3.1220988	5.370157	1.980107
K	4.229167	0.0000000	8.169367	0.6172819	11.647718	1.902806
N	17.058099	8.1693666	0.000000	8.5043830	17.864808	10.804008
P	3.122099	0.6172819	8.504383	0.0000000	8.432678	2.220191
Q	5.370157	11.6477178	17.864808	8.4326779	0.000000	9.486752
R	1.980107	1.9028059	10.804008	2.2201911	9.486752	0.000000

Question 2.3

We use from page 334

Theorem 5.12

Using the significance level α, the critical area for a test of the hypothesis $\mu_1 = \mu_2$ against all alternatives becomes

$$C = \left\{ x_{11}, \cdots, x_{1n_1}, x_{21}, \cdots, x_{2n_2} \;\middle|\; \frac{n_1+n_2-p-1}{p(n_1+n_2-2)} \cdot \frac{n_1 n_2}{n_1+n_2} d^2 > F(p, n_1+n_2-p-1)_{1-\alpha} \right\}$$

Here d^2 is the observed value of D^2.

Or equivalently from page 284

Theorem 4.9

We use the same notation as given above. Now, let

$$T^2 = \frac{nm}{n+m}(\bar{X}-\bar{Y})^T \mathbf{S}^{-1}(\bar{X}-\bar{Y}).$$

Then the critical region for a test of H_0 against H_1 at level α is equal to

$$C = \{x_1,\ldots,x_n,y_1,\ldots,y_m \mid \frac{n+m-p-1}{(n+m-2)p}t^2 > \mathrm{F}(p,n+m-p-1)_{1-\alpha}\}$$

Here t^2 is the observed value of T^2.

In the output, we can find the relevant information:

Total Sample Size	76	DF Total	75
Variables	8	DF Within Classes	70
Classes	6	DF Between Classes	5

Class Level Information

mfr	Variable Name	Frequency	Weight	Proportion	Prior Probability
G	G	22	22.0000	0.289474	0.166667
K	K	23	23.0000	0.302632	0.166667
N	N	6	6.0000	0.078947	0.166667
P	P	9	9.0000	0.118421	0.166667
Q	Q	8	8.0000	0.105263	0.166667
R	R	8	8.0000	0.105263	0.166667

Generalized Squared Distance to mfr

From mfr	G	K	N	P	Q	R
G	0	4.22917	17.05810	3.12210	5.37016	1.98011
K	4.22917	0	8.16937	0.61728	11.64772	1.90281
N	17.05810	8.16937	0	8.50438	17.86481	10.80401
P	3.12210	0.61728	8.50438	0	8.43268	2.22019
Q	5.37016	11.64772	17.86481	8.43268	0	9.48675
R	1.98011	1.90281	10.80401	2.22019	9.48675	0

In the notation of theorem 4.9 we have

Number of observations from manufacturer K: m=23

Number of observations from manufacturer R: n=8

Number of variable: p=8

We insert and get $f = 1.90281 \cdot \frac{23\cdot 8}{23+8} \cdot \frac{23+8-8-1}{(23+8-2)8} = 1.0710$

Which we compare against F(8,22) and get the p-value 0.4177

R code:

```
# We take the lowest value and perform the Hotelling's T^2 test
m =23 # Number of observations from manufacturer K
```

```
n = 8 # Number of observations from manufacturer R
p=8 #Number of variables
t_squared <- maha["R", "K"]
F_statistic <- ((n*m)/(n+m))*((n+m-p-1)/((n+m-2)*p))*t_squared
# Calculate the degrees of freedom for the F-distribution
df1 <- p
df2 <- n+m-p-1

# Calculate the p-value using the CDF of the F-distribution
p_value <- 1 - pf(F_statistic, df1, df2)

# Print the F-statistic and the p-value
cat("F-statistic:", F_statistic, "\n")
cat("P-value:", p_value, "\n")
```

R output:

F-statistic: 1.07099
P-value: 0.4177429

Question 2.4

We know need to run some additional code, to get the Mahalanobis' distance for the reduced set of variables

```
proc discrim data=breakfast pool=yes;
class mfr;
var protein fat sodium fiber carbo sugars potass calories;
run;
proc discrim data=breakfast pool=yes;
class mfr;
var fat sodium fiber carbo sugars calories;
run;
```

R code for reduced model:

```
variables_reduced <- c("calories", "fat", "sodium", "fiber", "carbo", "sugars")
# Perform Linear Discriminant Analysis
z_reduced <- lda(class ~ ., data = breakfast[, c("class", variables_reduced)],prior=prior)
library(Rfast)
pcov_reduced  <- pooled.cov(as.matrix(breakfast[,variables_reduced]),breakfast$mfr)
Means_reduced <- as.matrix(z_reduced$means)
invCov <- solve(pcov_reduced)

# Extract unique levels from breakfast$mfr
unique_levels <- levels(breakfast$mfr)
```

```
num_col <- length(unique(breakfast$mfr))

# Create an empty matrix to store the Mahalanobis distances with the equal priors ###############
maha_reduced <- matrix(c(rep(0, num_col^2)), ncol = num_col)

# Define the names for rows and columns (assuming unique_levels contains the names)
rownames(maha_reduced) <- unique_levels
colnames(maha_reduced) <- unique_levels

for (i in 1:num_col) {
  for (j in 1:num_col) {
    mu <- Means_reduced[i, ] - Means_reduced[j, ]
    maha_reduced[i, j] <- mu %*% invCov %*% mu
  }
}

# squared Mahalanobis distances between the 6 mfr types.
# maha    : Assuming equal priors

print("Generalized Squared Distance to species (equal priors):")
maha_reduced
```

We use from page 358

Theorem 5.21

The critical region for testing the hypothesis that the last $p - q$ variables do not contribute to the discrimination between the populations π_1 and π_2, i.e. the hypothesis that $\Delta^2_{(2|1)} = 0$ against all alternatives is

$$\left\{x_{11}, \cdots, x_{2n_2} \;\middle|\; \frac{n_1+n_2-p-1}{p-q} \frac{d^2-d_1^2}{(n_1+n_2)(n_1+n_2-2)/(n_1 n_2)+d_1^2} > F(p-q, n_1+n_2-p-1)_{1-\alpha}\right\}$$

Here d^2 and d_1^2 are the observed values of D^2 and D_1^2.

We find it the output

Generalized Squared Distance to mfr						
From mfr	G	K	N	P	Q	R
G	0	4.22917	17.05810	3.12210	5.37016	1.98011
K	4.22917	0	8.16937	0.61728	11.64772	1.90281
N	17.05810	8.16937	0	8.50438	17.86481	10.80401
P	3.12210	0.61728	8.50438	0	8.43268	2.22019
Q	5.37016	11.64772	17.86481	8.43268	0	9.48675
R	1.98011	1.90281	10.80401	2.22019	9.48675	0

Generalized Squared Distance to mfr						
From mfr	G	K	N	P	Q	R
G	0	2.72664	14.90960	2.42976	5.34844	1.36745
K	2.72664	0	8.03388	0.37206	9.79994	1.65055
N	14.90960	8.03388	0	8.09294	15.35592	10.33616
P	2.42976	0.37206	8.09294	0	7.54083	2.21568
Q	5.34844	9.79994	15.35592	7.54083	0	8.67643
R	1.36745	1.65055	10.33616	2.21568	8.67643	0

We have:

```
n1 = 8
n2 = 23
p = 8
q = 6
d1 = 1.90281
d2 = 1.65055
```

We insert and get an f-value

```
f = ((n1+n2-p-1)/(p-q)) * ((d1-d2)/ ( (n1+n2)*(n1+n2-2) / (n1*n2) + d2 ) )
 = 0.4245,
```

which we consider in F(2,22) and get a P-value of 0.6593

R code output reduced model:

```
          G         K         N         P         Q         R
G  0.000000 2.7266388 14.909599 2.4297614  5.348437  1.367450
K  2.726639 0.0000000  8.033879 0.3720592  9.799940  1.650553
N 14.909599 8.0338786  0.000000 8.0929394 15.355922 10.336160
P  2.429761 0.3720592  8.092939 0.0000000  7.540826  2.215683
Q  5.348437 9.7999398 15.355922 7.5408257  0.000000  8.676428
R  1.367450 1.6505529 10.336160 2.2156832  8.676428  0.000000
```

R output initial model from previous question:

	G	K	N	P	Q	R
G	0.000000	4.2291667	17.058099	3.1220988	5.370157	1.980107
K	4.229167	0.0000000	8.169367	0.6172819	11.647718	1.902806
N	17.058099	8.1693666	0.000000	8.5043830	17.864808	10.804008
P	3.122099	0.6172819	8.504383	0.0000000	8.432678	2.220191
Q	5.370157	11.6477178	17.864808	8.4326779	0.000000	9.486752
R	1.980107	1.9028059	10.804008	2.2201911	9.486752	0.000000

R code for F-statistic test:

```
######## Perform the statistical test ########################
n1 = 8
n2 = 23
p = 8
q = 6
d1 = 1.90281
d2 = 1.65055

#f-value
f_statistic_reduced = ((n1+n2-p-1)/(p-q)) * ((d1-d2)/ ( (n1+n2)*(n1+n2-2) / (n1*n2) + d2 ) )


# Calculate the degrees of freedom for the F-distribution
df1 <- p-q
df2 <- n1+n2-p-1

# Calculate the p-value using the CDF of the F-distribution
p_value <- 1 - pf(f_statistic_reduced, df1, df2)

# Print the F-statistic and the p-value
cat("F-statistic:", f_statistic_reduced, "\n")
cat("P-value:", p_value, "\n")
```

R output:

F-statistic: 0.4253
P-value: 0.6593

[0.6, 0.8[

Problem 3

Question 3.1

The correlation matrix, since the variables are on a very different scale.

Question 3.2

We can either run PRINCOMP or FACTOR in SAS, to get the eigenvalues.

We use from page 375, Theorem 6.8

If we instead are using the estimated *correlation matrix* $\hat{\mathbf{R}}$ we get the criterion

$$Z_2 = -n \log \frac{\det \hat{\mathbf{R}}}{\hat{\lambda}_1 \cdot \ldots \cdot \hat{\lambda}_m \cdot \hat{\lambda}^{k-m}} = -n \log \frac{\hat{\lambda}_{m+1} \cdot \ldots \cdot \hat{\lambda}_k}{\hat{\lambda}_*^{k-m}},$$

where

$$\hat{\lambda}_* = (k - \hat{\lambda}_1 - \ldots - \hat{\lambda}_m)/(k-m) = (\hat{\lambda}_{m+1} + \ldots + \hat{\lambda}_k)/(k-m).$$

The critical region for a test at level α becomes approximately equal to

$$\{x_1, \ldots, x_n | z_2 > \chi^2(\frac{1}{2}(k-m+2)(k-m-1))_{1-\alpha}\}.$$

However, it should be noted that this approximation is far worse than the corresponding approximation for the variance-covariance matrix.

The FACTOR Procedure
Initial Factor Method: Principal Components

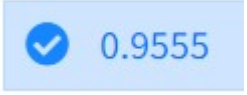
Prior Communality Estimates: ONE

Eigenvalues of the Correlation Matrix: Total = 8 Average = 1

	Eigenvalue	Difference	Proportion	Cumulative
1	2.65959651	0.67008608	0.3324	0.3324
2	1.98951043	0.57006752	0.2487	0.5811
3	1.41944291	0.53162008	0.1774	0.7586
4	0.88782283	0.34813403	0.1110	0.8695
5	0.53968880	0.15970917	0.0675	0.9370
6	0.37997963	0.31107246	0.0475	0.9845
7	0.06890717	0.01385543	0.0086	0.9931
8	0.05505174		0.0069	1.0000

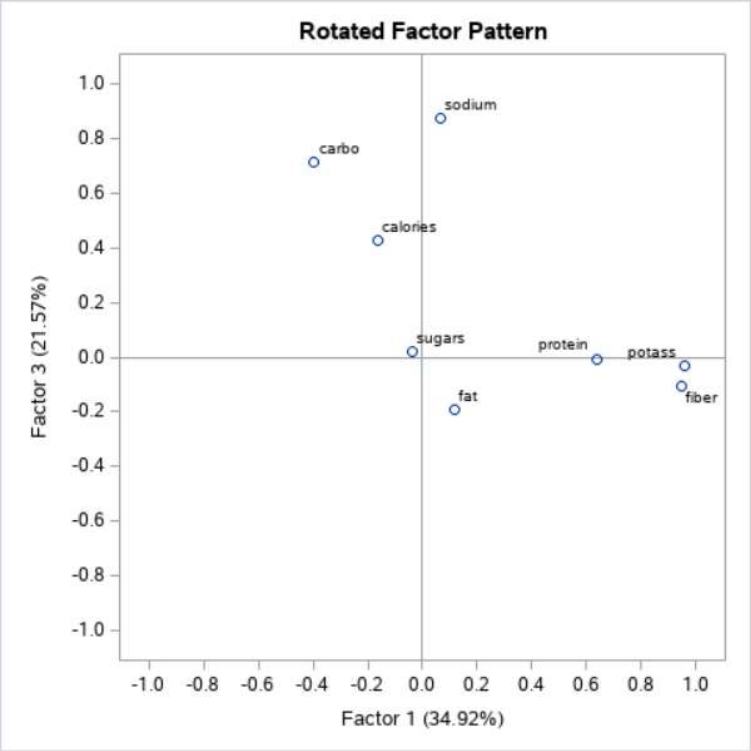
We have the last to eigenvalues from the output, k=8, m=6, and n=76. We plug in

```
k = 8
m = 6
n = 76
eig7 = 0.06890717
eig8 = 0.05505174
lambda = (eig7+eig8)/(k-m)
z2 = -n*log((eig7*eig8)/(lambda^(k-m))) = 0.9555
```



Question 3.3

Consider the factor patterns – either the table or the plots.



Rotated Factor 1 is mainly a contrast between positive loadings of *fiber*, *potass*, and *protein* vs. *carbo*.
Rotated Factor 3 is mainly a weighting of *sodium*, *carbo*, and *calories*.

Question 3.4

"We again consider a factor analysis - as described in the problem - with 4 factors*. The uniqueness of* protein *is:"*

We find in the output

Rotated Factor Pattern				
	Factor1	Factor2	Factor3	Factor4
protein	0.63903	0.34998	-0.01125	-0.54056
fat	0.11557	0.87899	-0.19110	0.08489
sodium	0.06443	0.03574	0.87364	0.13687
fiber	0.94843	-0.14699	-0.10602	-0.06173
carbo	-0.39805	-0.08910	0.71036	-0.40915
sugars	-0.03852	0.32858	0.02119	0.90440
potass	0.95801	0.07735	-0.03346	0.04029
calories	-0.16104	0.77323	0.42776	0.28348

Variance Explained by Each Factor			
Factor1	Factor2	Factor3	Factor4
2.4290319	1.6377759	1.5002884	1.3892765

Final Communality Estimates: Total = 6.956373							
protein	fat	sodium	fiber	carbo	sugars	potass	calories
0.82316829	0.82971237	0.78740839	0.93618199	0.83840423	0.92783529	0.92650616	0.88715595

This is simple 1 minus the communality. The communality is given by the sum of squared factor loadings and we thus get:

$$1 - 0.63903^2 - 0.34998^2 - 0.01125^2 - 0.54056^2$$

Question 3.5

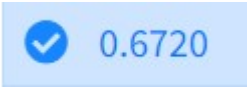
We find the loadings in the output and square them

Rotated Factor Pattern				
	Factor1	Factor2	Factor3	Factor4
protein	0.63903	0.34998	-0.01125	-0.54056
fat	0.11557	0.87899	-0.19110	0.08489
sodium	0.06443	0.03574	0.87364	0.13687
fiber	0.94843	-0.14699	-0.10602	-0.06173
carbo	-0.39805	-0.08910	0.71036	-0.40915
sugars	-0.03852	0.32858	0.02119	0.90440
potass	0.95801	0.07735	-0.03346	0.04029
calories	-0.16104	0.77323	0.42776	0.28348

Variance Explained by Each Factor			
Factor1	Factor2	Factor3	Factor4
2.4290319	1.6377759	1.5002884	1.3892765

Final Communality Estimates: Total = 6.956373							
protein	fat	sodium	fiber	carbo	sugars	potass	calories
0.82316829	0.82971237	0.78740839	0.93618199	0.83840423	0.92783529	0.92650616	0.88715595

$$(0.71036)^2 + (-0.40915)^2 = 0.5046 + 0.1674 = 0.6720$$



Problem 4

We run the following code

```
* Problem 4;
proc reg data=breakfast plots(label) = all;
model rating = calories protein fat sodium fiber carbo sugars potass/r
influence vif tol;
run;
```

Question 4.1

This is just R^2 – and the expression 1-SSerror/SStotal, is just another way to write it. We find it in

Remark 3.6 Squared multiple correlation

For the **squared multiple correlation coefficient** we obtain (skipping the index on X so that X is the k dimensional random variable not to be mixed up with the $n \times k$ dimensional data matrix)

$$\rho_{y|x} = \frac{\sqrt{\mathrm{V}(Y|\boldsymbol{X})}}{\sqrt{\mathrm{V}(Y)}} = \frac{\sqrt{\sigma_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\sigma_{xy}}}{\sigma_y}$$

giving

$$\rho_{y|x}^2 = \frac{\mathrm{V}(Y) - \mathrm{V}(Y|\boldsymbol{X})}{\mathrm{V}(Y)} = \frac{\sigma_y^2 - \sigma_{y|x}^2}{\sigma_y^2}$$

where

$$\sigma_{y|x}^2 = \sigma_y^2 - \sigma_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\sigma_{xy}$$

The empirical version becomes

$$\widehat{\rho}_{y|x}^2 = \frac{\widehat{\mathrm{V}}(Y) - \widehat{\mathrm{V}}(Y|\boldsymbol{X})}{\widehat{\mathrm{V}}(Y)}$$

$$= \frac{\sum (Y_i - \overline{Y})^2 - \boldsymbol{R}_y^T \boldsymbol{R}_y}{\sum (Y_i - \overline{Y})^2}$$

$$= \frac{(\boldsymbol{Y} - \mathbf{1}\overline{Y})^T (\boldsymbol{Y} - \mathbf{1}\overline{Y}) - \boldsymbol{R}_y^T \boldsymbol{R}_y}{(\boldsymbol{Y} - \mathbf{1}\overline{Y})^T (\boldsymbol{Y} - \mathbf{1}\overline{Y})}$$

$$= \frac{SS_{Tot(y)} - SS_{Res(y|x)}}{SS_{Tot(y)}}$$

We find in the output

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	8	14766	1845.79844	1546.25	<.0001
Error	67	79.97979	1.19373		
Corrected Total	75	14846			

Root MSE	1.09258	R-Square	0.9946
Dependent Mean	42.50537	Adj R-Sq	0.9940
Coeff Var	2.57045		

R code:

```
# Fit a linear regression model
Model <- lm(rating ~ calories + protein + fat + sodium + fiber + carbo + sugars + potass, data = breakfast)

# Calculate the sum of squares for the model
SS_total <- sum((breakfast$rating - mean(breakfast$rating))^2)

# Calculate the sum of squares for the error
SS_error <- sum((breakfast$rating - predict(Model))^2)

# Print SS model and SS error
cat("Sum of Squares (SS) total:", SS_total, "\n")
cat("Sum of Squares (SS) Error:", SS_error, "\n")
```

R output:

Sum of Squares (SS) Total: 14846.37
Sum of Squares (SS) Error: 79.97979

$$1 - \frac{79.9798}{14846}$$

Note: Error in the exam and only the option with 1-SS_error/SS_model is present. However, numerically they are almost identical.

Question 4.2

We use from page 44

Theorem 1.45

Let $R = \hat{\rho}_{y_i|x}$ be the empirical multiple correlation coefficient between Y_i and $\boldsymbol{X} = (Z_{m+1}, \ldots, Z_p)$ based upon n observations. Then

$$\frac{R^2}{1-R^2} \cdot \frac{n-(p-m)-1}{p-m} \sim \mathrm{F}(p-m, n-(p-m)-1),$$

if $\rho_{y_i|x} = \rho_{y_i|z_{m+1},\ldots,p} = 0$.

Remark 1.46

The number $p - m$ is equal to the number of variables in $\boldsymbol{X}$, i.e. the number of variables we condition on.

We have 76 observation: n=76

We have 8 independent variables in the model, (p – m) = 8

We insert: F(p-m, n – (p - m) - 1) = F(8, 76 - 8 - 1) = F(8, 67)

F(8,67)

Question 4.3

In SAS we specify VIF and TOL in the model statement

We have from page 208

Definition 3.15

We define the *tolerance (TOL)* and the *variance inflation (VIF)* as

$$\begin{aligned} \text{TOL}_i &= 1 - R^2\,(x_i|\text{all other } x\text{-variables}) \\ \text{VIF}_i &= \frac{1}{\text{TOL}_i} \end{aligned}$$

As a rule of thumb, TOL < 0.1 or equivalently VIF > 10 indicates a multi-collinearity problem.

We find in the output

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Tolerance	Variance Inflation
Intercept	1	55.98943	0.91576	61.14	<.0001	.	0
calories	1	-0.22869	0.01599	-14.30	<.0001	0.16213	6.16792
protein	1	3.20763	0.18475	17.36	<.0001	0.39320	2.54321
fat	1	-1.60147	0.20015	-8.00	<.0001	0.38708	2.58347
sodium	1	-0.05802	0.00172	-33.65	<.0001	0.79005	1.26575
fiber	1	3.41506	0.15839	21.56	<.0001	0.11143	8.97427
carbo	1	1.03646	0.06078	17.05	<.0001	0.23258	4.29953
sugars	1	-0.76763	0.06452	-11.90	<.0001	0.19301	5.18112
potass	1	-0.03452	0.00523	-6.60	<.0001	0.11292	8.85587

R code:

```
# Load the required library
library(car)

# Fit a linear regression model
```

```
Model <- lm(rating ~ calories + protein + fat + sodium + fiber + carbo + sugars + potass, data = breakfast)

# Calculate VIF for each variable
vif_values <- vif(Model)

# Create a data frame with variable names and VIF values
independent_variables <- names(coefficients(Model)[-1]) # Exclude the intercept
vif_df <- data.frame(Variable = independent_variables, VIF = vif_values)

# Calculate Tolerance as the reciprocal of VIF
vif_df$Tolerance <- 1 / vif_df$VIF

# Print the VIF and Tolerance values
print(vif_df)
```

R output:

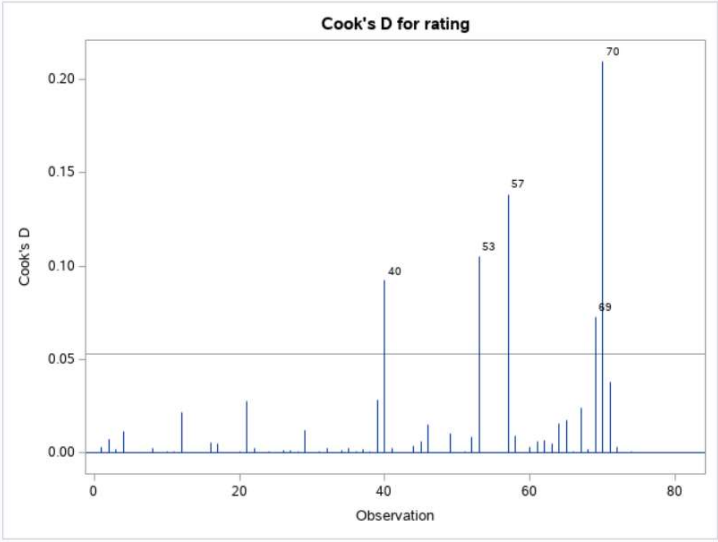
```
         Variable      VIF Tolerance
calories calories 6.167919 0.1621292
protein   protein 2.543206 0.3932045
fat           fat 2.583475 0.3870756
sodium     sodium 1.265749 0.7900461
fiber       fiber 8.974267 0.1114297
carbo       carbo 4.299526 0.2325838
sugars     sugars 5.181118 0.1930085
potass     potass 8.855874 0.1129194
```

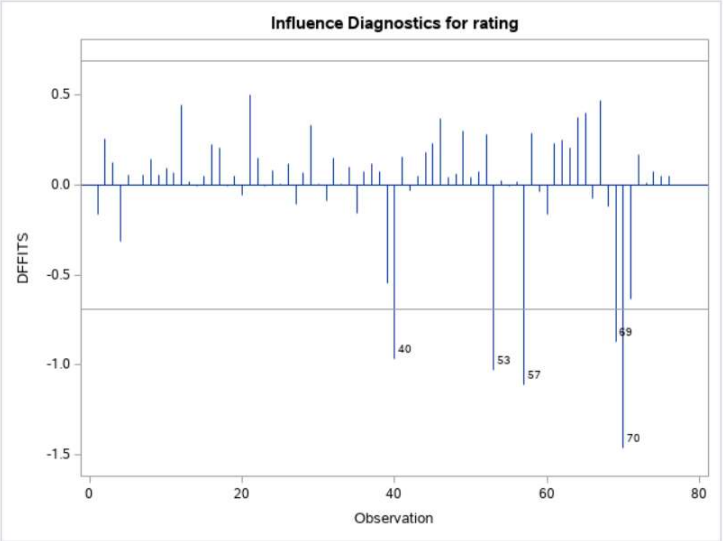
Since neither the TOL nor VIF are outside the boundaries, and since they *do not* give complementary information, the answer is:

> The lowest tolerance found is 0.111 and the largest VIF is 8.97. We are thus within the rules of thumb and multicolinearity does not seem to be a problem.

Question 4.4

We look for Cook's D and DFFITS in the output:



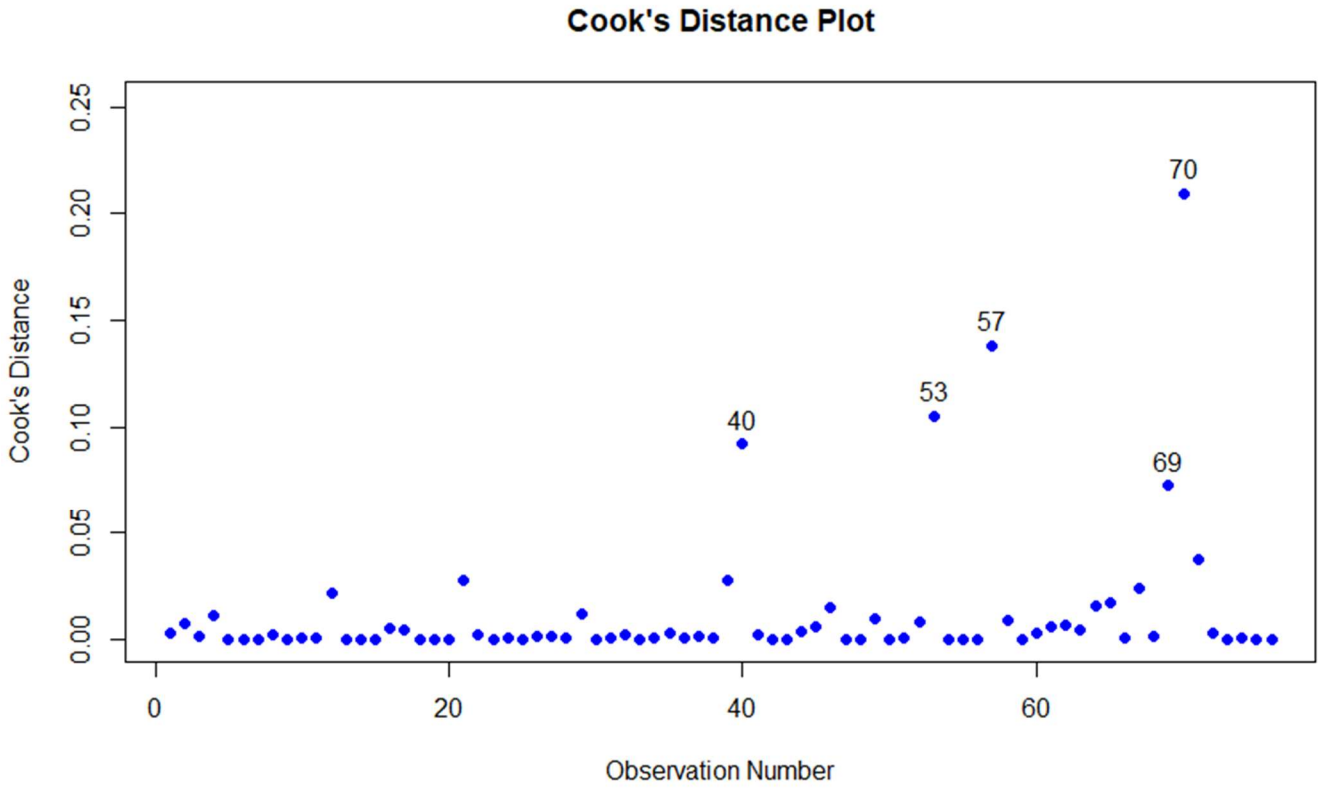


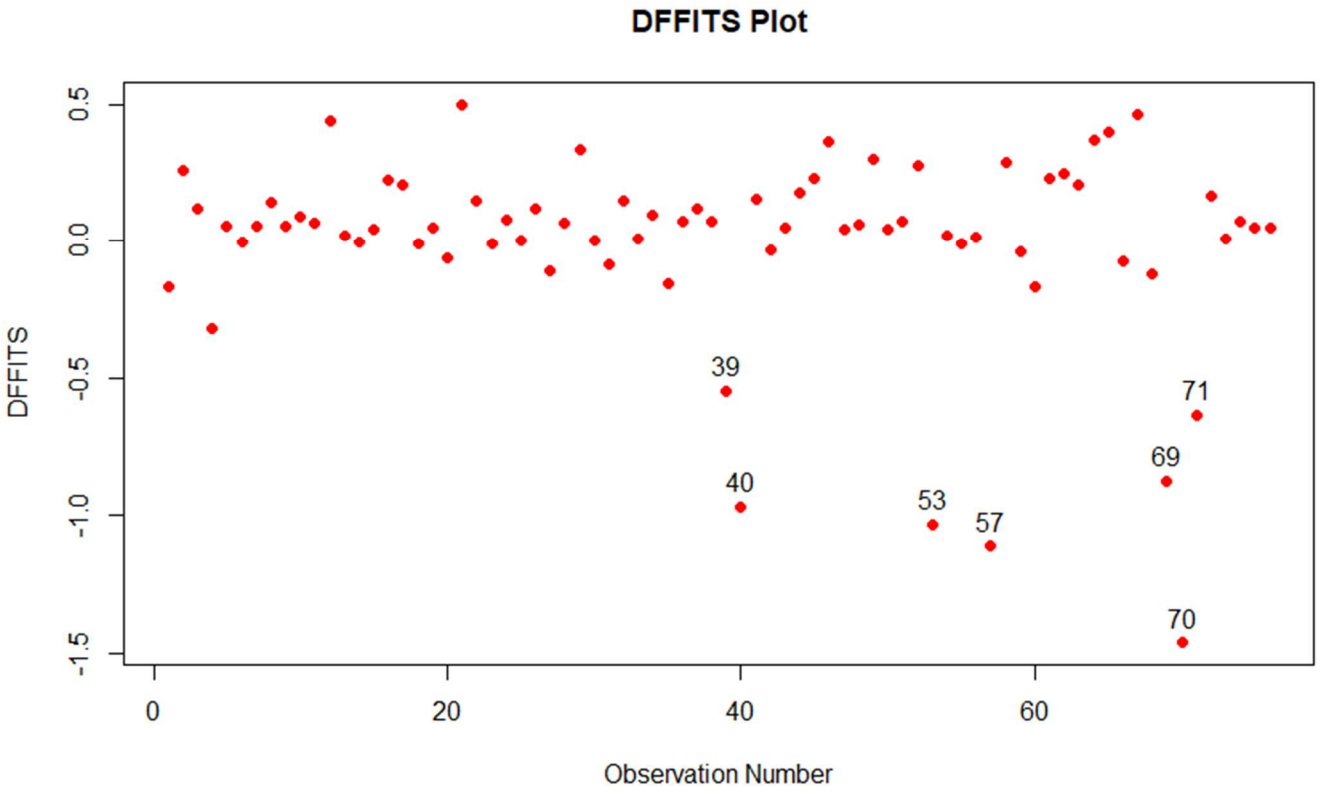
We see observation 70 in both cases.

R code:

```
Obs <- 1:length(breakfast$rating)
CookD <- round(cooks.distance(Model), 5)
DFFITS <- round(dffits(Model), 5)
Stats  <- data.frame(Obs,CookD,DFFITS)
Stats

# Cook's D plot
# Calculate the maximum Cook's Distance value
max_cook <- max(Stats$CookD)

# Create a plot for Cook's Distance with adjusted y-axis limits
plot(Stats$Obs, Stats$CookD, type = "p", pch = 19, col = "blue",
    xlab = "Observation Number", ylab = "Cook's Distance",
    main = "Cook's Distance Plot",
    ylim = c(0, 1.20 * max_cook))

# Identify and label observations with Cook's Distance > 0.05
high_cook_obs <- Stats$Obs[Stats$CookD > 0.05]
text(high_cook_obs, Stats$CookD[Stats$CookD > 0.05], labels = high_cook_obs, pos = 3)

# Create a plot for DFFITS with adjusted y-axis limits
plot(Stats$Obs, Stats$DFFITS, type = "p", pch = 19, col = "red",
    xlab = "Observation Number", ylab = "DFFITS",
    main = "DFFITS Plot")

# Identify and label observations with  DFFITS < -0.5
low_dffits_obs <- Stats$Obs[(Stats$DFFITS) < -0.5]
text(low_dffits_obs, Stats$DFFITS[(Stats$DFFITS) < -0.5], labels = low_dffits_obs, pos = 3)
```

R output:

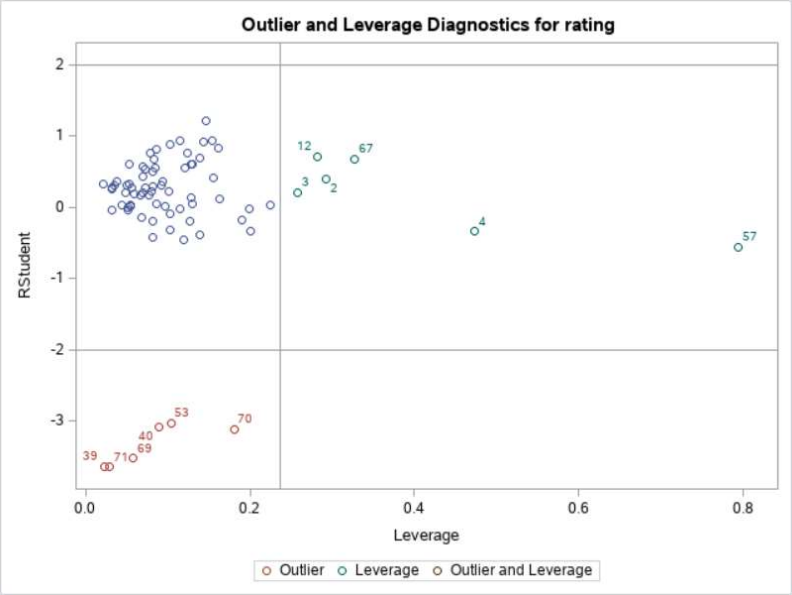

Cook's Distance Plot
Cook's Distance
0.00
0.05
0.10
0.15
0.20
0.25
0
20
40
60
Observation Number
70
57
53
40
69


DFFITS Plot
DFFITS
-1.5
-1.0
-0.5
0.0
0.5
0
20
40
60
Observation Number
39
40
53
57
69
71
70

70

Question 4.5

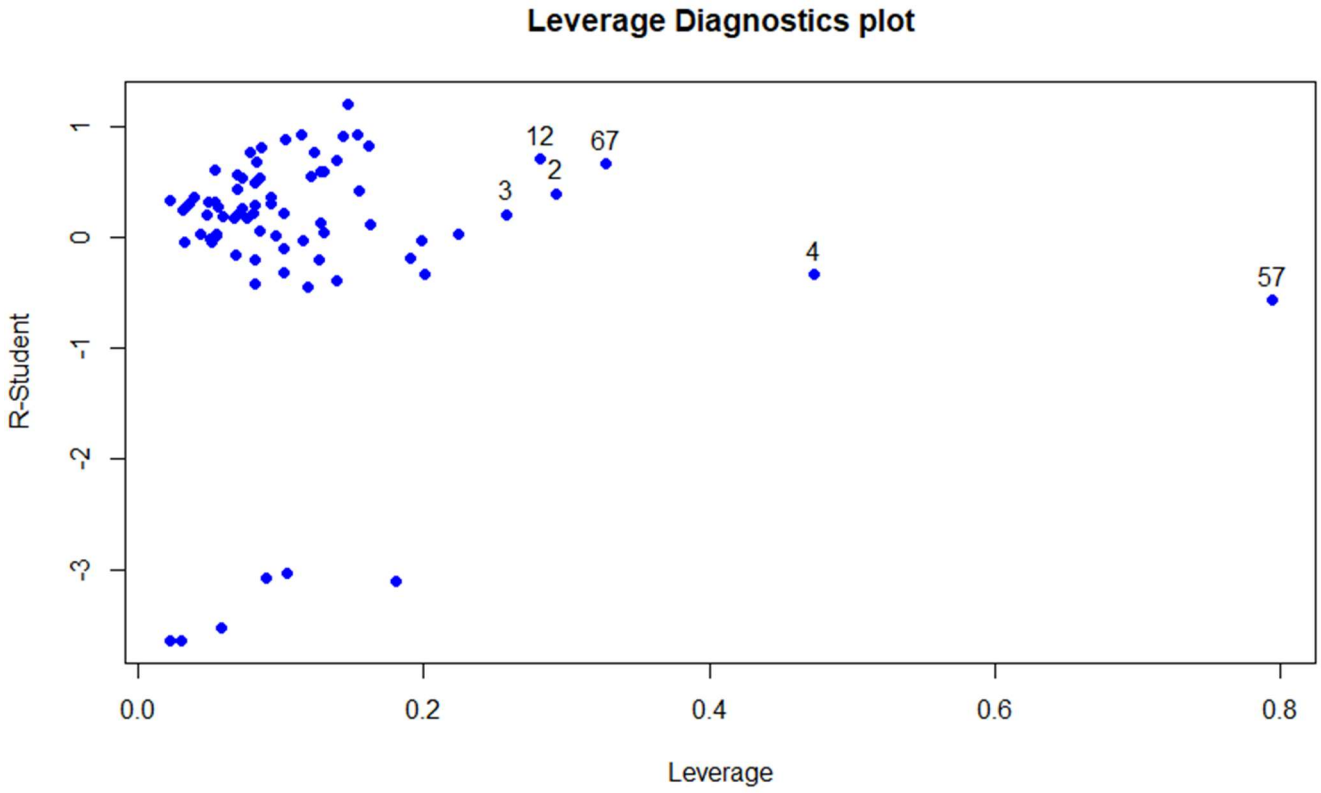
The largest potential to influence the model, is measured by the *leverage.* We find in the output.



R code:

```
# Calculate R student and leverage values
R_student <- rstudent(Model)
leverage <- hatvalues(Model)
Stats$Rstudent  <- R_student
Stats$leverage <- leverage
# Create the residuals vs. leverage plot
plot(leverage, R_student, type = "p", pch = 19, col = "blue",
    xlab = "Leverage", ylab = "R-Student",
    main = "Leverage Diagnostics plot")

# Identify and label observations with high leverage, high R-student
high_leverage_obs <- which(leverage > 2 * mean(leverage))  # Example leverage threshold
R_student_obs <- which(abs(R_student) > 2)         # Example rstudent
text(leverage[high_leverage_obs], R_student[high_leverage_obs], labels = high_leverage_obs, pos = 3)
```

R output:



57

Question 4.6

The F-value is not given directly in either SAR or R. However, we have from page 133

Remark 2.25

The test given in Theorem 2.23 is of course equivalent to the ordinary F-test presented in Theorem 2.21 for $k - r = 1$. This may be established directly using the fact that the square of a t-distributed random variable is F-distributed, mnemonically written

$$[t(f)]^2 = F(1, f).$$

The advantage by using Theorem 2.23 is that we with obvious modifications may test one-sided hypothesis like $H_0: \theta_{i_0} \leq c$ against $H_1: \theta_{i_0} > c$. This is not possible with the F-test.

I.e., a monotonic relationship between the t and F value, and we simply pick the smallest t-value.

We find in the output:

Parameter Estimates							
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Tolerance	Variance Inflation
Intercept	1	55.98943	0.91576	61.14	<.0001	.	0
calories	1	-0.22869	0.01599	-14.30	<.0001	0.16213	6.16792
protein	1	3.20763	0.18475	17.36	<.0001	0.39320	2.54321
fat	1	-1.60147	0.20015	-8.00	<.0001	0.38708	2.58347
sodium	1	-0.05802	0.00172	-33.65	<.0001	0.79005	1.26575
fiber	1	3.41506	0.15839	21.56	<.0001	0.11143	8.97427
carbo	1	1.03646	0.06078	17.05	<.0001	0.23258	4.29953
sugars	1	-0.76763	0.06452	-11.90	<.0001	0.19301	5.18112
potass	1	-0.03452	0.00523	-6.60	<.0001	0.11292	8.85587

R code:

summary(Model)

R output:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 55.989429   0.915760  61.140  < 2e-16 ***
calories    -0.228688   0.015988 -14.304  < 2e-16 ***
protein      3.207625   0.184747  17.362  < 2e-16 ***
fat         -1.601468   0.200146  -8.002 2.39e-11 ***
sodium      -0.058024   0.001724 -33.654  < 2e-16 ***
fiber        3.415056   0.158390  21.561  < 2e-16 ***
carbo        1.036456   0.060776  17.054  < 2e-16 ***
sugars      -0.767629   0.064515 -11.898  < 2e-16 ***
potass      -0.034522   0.005232  -6.599 7.87e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.093 on 67 degrees of freedom
Multiple R-squared:  0.9946,    Adjusted R-squared:  0.994
F-statistic:  1546 on 8 and 67 DF,  p-value: < 2.2e-16
```

Question 4.7

In the model the "rating" is the dependent value. We can thus simply look at the parameter estimates. Large positive estimates would indicate that increasing that quantity, would lead to a better breakfast – at least one with a higher rating.

Parameter Estimates							
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Tolerance	Variance Inflation
Intercept	1	55.98943	0.91576	61.14	<.0001	.	0
calories	1	-0.22869	0.01599	-14.30	<.0001	0.16213	6.16792
protein	1	3.20763	0.18475	17.36	<.0001	0.39320	2.54321
fat	1	-1.60147	0.20015	-8.00	<.0001	0.38708	2.58347
sodium	1	-0.05802	0.00172	-33.65	<.0001	0.79005	1.26575
fiber	1	3.41506	0.15839	21.56	<.0001	0.11143	8.97427
carbo	1	1.03646	0.06078	17.05	<.0001	0.23258	4.29953
sugars	1	-0.76763	0.06452	-11.90	<.0001	0.19301	5.18112
potass	1	-0.03452	0.00523	-6.60	<.0001	0.11292	8.85587

We have a breakfast with a large amount of fiber and protein and some carbohydrates. At the same time, it should have a minimal amount of fat and sugar.

R output:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 55.989429   0.915760  61.140  < 2e-16 ***
calories    -0.228688   0.015988 -14.304  < 2e-16 ***
protein      3.207625   0.184747  17.362  < 2e-16 ***
fat         -1.601468   0.200146  -8.002 2.39e-11 ***
sodium      -0.058024   0.001724 -33.654  < 2e-16 ***
fiber        3.415056   0.158390  21.561  < 2e-16 ***
carbo        1.036456   0.060776  17.054  < 2e-16 ***
sugars      -0.767629   0.064515 -11.898  < 2e-16 ***
potass      -0.034522   0.005232  -6.599 7.87e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.093 on 67 degrees of freedom
Multiple R-squared:  0.9946,    Adjusted R-squared:  0.994
F-statistic:  1546 on 8 and 67 DF,  p-value: < 2.2e-16
```

Problem 5

The key insight is how to set up the model matrix. We have

	columns	
rows	$E(Y_1) = \mu + \alpha$	$E(Y_2) = \mu - \alpha$
	$\mathrm{E}(Y_3) = \mu - \alpha$	$E(Y_4) = \mu + \alpha + \beta$

That leads to a model matrix of

$$\boldsymbol{x} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & -1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Further $\boldsymbol{x}^T\boldsymbol{x} = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, the inverse is given in the problem, together with the least squares estimate.

$$\begin{bmatrix} 4 & 0 & 1 \\ 0 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}^{-1} = \frac{1}{8}\begin{bmatrix} 3 & 1 & -4 \\ 1 & 3 & -4 \\ -4 & -4 & 16 \end{bmatrix}$$

$$\frac{1}{8}\begin{bmatrix} 3 & 1 & -4 \\ 1 & 3 & -4 \\ -4 & -4 & 16 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 2 & 1 & 1 & 0 \\ 2 & -1 & -1 & 0 \\ -4 & 0 & 0 & 4 \end{bmatrix}$$

Question 5.1

Can be read from the least squares solution, first line pertaining to $\hat{\mu}$

$$\frac{1}{4}\begin{bmatrix} 2 & 1 & 1 & 0 \\ 2 & -1 & -1 & 0 \\ -4 & 0 & 0 & 4 \end{bmatrix}$$

$\frac{1}{4}[2 \quad 1 \quad 1 \quad 0]$

Question 5.2

From page 103, Theorem 2.3 we have:

$$\begin{aligned} \mathrm{E}(\hat{\boldsymbol{\theta}}) &= \boldsymbol{\theta} \\ \mathrm{D}(\hat{\boldsymbol{\theta}}) &= \sigma^2(\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x})^{-1}. \end{aligned}$$

Since we have independent observations with variance σ^2, we get

$$D\left(\begin{bmatrix} \hat{\mu} \\ \hat{\alpha} \\ \hat{\beta} \end{bmatrix}\right) = \sigma^2\frac{1}{8}\begin{bmatrix} 3 & 1 & -4 \\ 1 & 3 & -4 \\ -4 & -4 & 16 \end{bmatrix},$$

Given a variance for $\hat{\mu}$

$\frac{3}{8}\sigma^2$

Question 5.3

We again read directly from the least squares solution, first third line pertaining to $\hat{\beta}$

$$\frac{1}{4}\begin{bmatrix} 2 & 1 & 1 & 0 \\ 2 & -1 & -1 & 0 \\ -4 & 0 & 0 & 4 \end{bmatrix}$$

$[-1 \quad 0 \quad 0 \quad 1]$

Question 5.4

From page 103, Theorem 2.3 we have:

$$\begin{aligned} \mathrm{E}(\hat{\boldsymbol{\theta}}) &= \boldsymbol{\theta} \\ \mathrm{D}(\hat{\boldsymbol{\theta}}) &= \sigma^2(\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x})^{-1}. \end{aligned}$$

Since we have independent observations with variance σ^2, we get

$$D\left(\begin{bmatrix}\hat{\mu}\\ \hat{\alpha}\\ \hat{\beta}\end{bmatrix}\right) = \sigma^2\frac{1}{8}\begin{bmatrix}3 & 1 & -4\\ 1 & 3 & -4\\ -4 & -4 & 16\end{bmatrix},$$

Giving us the correlation $corr(\hat{\mu},\hat{\beta})$

$$Corr(\hat{\mu},\hat{\beta}) = \frac{-\frac{1}{2}\sigma^2}{\sqrt{\frac{3}{8}\sigma^2}\sqrt{\frac{16}{8}\sigma^2}} = -\frac{1}{\sqrt{3}} = -\frac{\sqrt{3}}{3},$$

$-\frac{\sqrt{3}}{3}$

Question 5.5

Since β =0, we modify our model matrix and estimate $\hat{\hat{\mu}}$

$$\hat{\hat{x}} = \begin{bmatrix}1 & 1\\ 1 & -1\\ 1 & -1\\ 1 & 1\end{bmatrix}$$

$$\left(\hat{\hat{x}}^T\hat{\hat{x}}\right)^{-1}\hat{\hat{x}}^T = \frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

We get from the first line

$\frac{1}{4}[1 \quad 1 \quad 1 \quad 1]$

Question 5.6

Here we can from page 11 use

Remark 1.10 Rules for computing moments of simple functions

$\mathrm{E}(a+bX) = a + b\,\mathrm{E}(X)$	$\mathrm{E}(\mathbf{A}+\mathbf{X}) = \mathbf{A} + \mathrm{E}(\mathbf{X})$
$\mathrm{E}(X+Y) = \mathrm{E}(X) + \mathrm{E}(Y)$	$\mathrm{E}(\mathbf{AX}) = \mathbf{A}\,\mathrm{E}(\mathbf{X})$
	$\mathrm{E}(\mathbf{XB}) = \mathrm{E}(\mathbf{X})\mathbf{B}$
	$\mathrm{E}(\mathbf{X}+\mathbf{Y}) = \mathrm{E}(\mathbf{X}) + \mathrm{E}(\mathbf{Y})$
$\mathrm{V}(a+bX) = b^2\,\mathrm{V}(X)$	$\mathrm{D}(\mathbf{b}+\mathbf{X}) = \mathrm{D}(\mathbf{X})$
$\mathrm{V}(X+Y) = \mathrm{V}(X) + \mathrm{V}(Y) + 2\mathrm{Cov}(X,Y)$	$\mathrm{D}(\mathbf{AX}) = \mathbf{A}\,\mathrm{D}(\mathbf{X})\mathbf{A}^T$
$= \mathrm{V}(X) + \mathrm{V}(Y)$	$\mathrm{D}(\mathbf{X}+\mathbf{Y}) = \mathrm{D}(\mathbf{X}) + \mathrm{D}(\mathbf{Y}) + \mathrm{C}(\mathbf{X},\mathbf{Y}) + \mathrm{C}(\mathbf{Y},\mathbf{X})$
iff X, Y independent	$= \mathrm{D}(\mathbf{X}) + \mathrm{D}(\mathbf{Y})$
	iff X, Y independent
$\mathrm{Cov}(X,X) = \mathrm{V}(X)$	$\mathrm{C}(\mathbf{X},\mathbf{X}) = \mathrm{D}(\mathbf{X})$
$\mathrm{Cov}(aX,bY) = ab\mathrm{Cov}(X,Y)$	$\mathrm{C}(\mathbf{X},\mathbf{Y}) = \mathrm{C}(\mathbf{Y},\mathbf{X})^T$
$\mathrm{Cov}(X+U,Y) = \mathrm{Cov}(X,Y) + \mathrm{Cov}(U,Y)$	$\mathrm{C}(\mathbf{AX},\mathbf{BY}) = \mathbf{A}\,\mathrm{C}(\mathbf{X},\mathbf{Y})\mathbf{B}^T$
$\mathrm{Cov}(X,Y+V) = \mathrm{Cov}(X,Y) + \mathrm{Cov}(X,V)$	$\mathrm{C}(\mathbf{X}+\mathbf{U},\mathbf{Y}) = \mathrm{C}(\mathbf{X},\mathbf{Y}) + \mathrm{C}(\mathbf{U},\mathbf{Y})$
	$\mathrm{C}(\mathbf{X},\mathbf{Y}+\mathbf{V}) = \mathrm{C}(\mathbf{X},\mathbf{Y}) + \mathrm{C}(\mathbf{X},\mathbf{V})$

$$Cov(A,B) = A\,C(X,X)B^T = A\,D(X)B^T = [0.5 \quad 0.25 \quad 0.25 \quad 0]\begin{bmatrix}\sigma^2 & 0 & 0 & 0\\ 0 & \sigma^2 & 0 & 0\\ 0 & 0 & \sigma^2 & 0\\ 0 & 0 & 0 & \sigma^2\end{bmatrix}\begin{bmatrix}0.25\\0.25\\0.25\\0.25\end{bmatrix} = \frac{\sigma^2}{4}$$

Now we need to normalize it, to get the correlation.

$$Corr(\hat{\mu},\hat{\mu}) = \frac{\frac{1}{4}}{\sqrt{V(\hat{\mu})V(\hat{\mu})}} = \frac{1}{4\sqrt{\frac{1}{4}\times\frac{3}{8}}} = \frac{\sqrt{2}}{\sqrt{3}} = \frac{\sqrt{6}}{3},$$

$\frac{\sqrt{6}}{3}$

Problem 6

Question 6.1

We have from page 290

If the observations are rearranged into a column vector

$$\underline{Y} = \mathrm{vc}(\mathbf{Y}) = \begin{bmatrix}\mathbf{Y}_{1|}\\ \vdots \\ \mathbf{Y}_{p|}\end{bmatrix},$$

we find from theorem 1.9, p. 10, that

$$\mathrm{D}(\mathbf{Y}) = \mathbf{\Sigma}\otimes\mathbf{I}_n = \begin{bmatrix}\sigma_1^2\mathbf{I}_n & \cdots & \sigma_{1p}\mathbf{I}_n\\ \vdots & & \vdots\\ \sigma_{p1}\mathbf{I}_n & \cdots & \sigma_p^2\mathbf{I}_n\end{bmatrix},$$

where $\mathbf{\Sigma}\otimes\mathbf{I}_n$ is the tensor product of $\mathbf{\Sigma}$ and $\mathbf{I}_n$, cf. section A.5.

We know $\mathbf{\Sigma}$, and since we have 2 observation, we thus get:

$$D(vc(\boldsymbol{Y})) = \boldsymbol{\Sigma} \otimes \mathbf{I}_{\boldsymbol{n}} = \begin{bmatrix} \sigma & \gamma & \rho \\ \gamma & \sigma & \eta \\ \rho & \eta & \sigma \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \sigma & 0 & \gamma & 0 & \rho & 0 \\ 0 & \sigma & 0 & \gamma & 0 & \rho \\ \gamma & 0 & \sigma & 0 & \eta & 0 \\ 0 & \gamma & 0 & \sigma & 0 & \eta \\ \rho & 0 & \eta & 0 & \sigma & 0 \\ 0 & \rho & 0 & \eta & 0 & \sigma \end{bmatrix}$$

Problem 7

Question 7.1

We know from page 29, that the squared correlation is the fraction of variance explained:

and the *squared coefficient of correlation represents the reduction in variance. i.e. the fraction of Y's variance, which can be explained by X,* since

$$\rho^2 = \frac{\mathrm{V}(Y) - \mathrm{V}(Y|X = x)}{\mathrm{V}(Y)}.$$

We thus take the canonical correlation and square it

$$0.3083^2 = 0.0950$$

0.0950

Question 7.2

We use from page 388

Remark 6.15

Consequently, we may test whether Y and X are independent by testing whether the first canonical correlation is 0. This may of course be done directly without the detour around the canonical correlations. If we estimate the dispersion parameters on the basis of n observations of Z, the test could be performed - as shown in section 4.4 - by investigating

$$\frac{|S|}{|S_{yy}|\,|S_{xx}|}$$

which is $U(p, q, n-1-q)$ distributed under the hypothesis of independence.

We further have from page 382 that

We consider a random variable

$$Z \sim N_{p+q}(\mu, \Sigma),$$

where $p \leq q$ and Z and the parameters have been partitioned as follows:

$$Z = \begin{bmatrix} Y \\ X \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_y \\ \mu_x \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{bmatrix}.$$

We then have p=6, q=11 and n=528, inserting U(6,11,528 – 1 - 11) = U(6,11,516)

U(6,11,516)

Question 7.3

For both the covariance and correlation matrix, we need p+1 samples, to account for the mean. The answer is thus p+1 = 11 +1 = 12

12