Stat-340/341 – Final Exam –

1 Part 1 - Multiple Choice

Enter your answers to the multiple choice questions on the provided bubble sheets. Each of the 20 multiple choice question is worth 1 mark – there is no correction for guessing. Be sure your student name and number are completed on the bubble sheets.

Students in Stat-341, should only answer questions 1-10 of the multiple choice section (those labelled as about the *R* language).

1. (*R*) Which is NOT a valid expression?

   (a) -3+5

   (b) -3 - - 5

   (c) c(3, "fred")

   (d) c("fred", 5) + 10

   (e) 3+TRUE

   Solution: (d)
   Option B - 14% chose. You need to distinguish between a unary minus and the binary minus operators.
   Option D - 43% chose.
   Option E - 39% chose. *R* will convert *TRUE* to a 1 before doing the addition.

2. (*R*) The data values on blood pressure readings are stored in the file *bp.csv*.

```
Name  ,     Sex  ,  Year   ,  BP
C     ,     f    ,  2009   ,  120
C     ,     0    ,  2010   ,  130
D     ,     1    ,  NA     ,  140
M     ,     0    ,  2011   ,  140
M     ,     m    ,  2012   ,  150
```

   This data was read using

```
my.data <- read.csv(bp.csv, header=TRUE, strip.white=TRUE)
```

   Which of the following is NOT correct?

(a)
```
> my.data[1,]
  Name Sex Year  BP
1    C   f 2009 120
```

(b)
```
> my.data[,1]
[1] C C D M M
```

(c)
```
> mean(my.data[,"BP"])
[1] 136
```

(d)
```
> my.data[,"Sex"]=='f'
[1]  TRUE FALSE FALSE FALSE FALSE
```

(e)
```
> sum(my.data[,"Year"],na.rm=TRUE)
[1] NA
```

Solution: (e)
Option E - 84% chose.

3. (*R*) Which of the following is correct about the standard error of a mean.

   (a) It measures the variation of individual items in the population when repeated samples are taken.

   (b) It measures how variable the population mean is when repeated samples are taken.

   (c) It measures the variation in the sample mean when a new sample is taken from the population.

   (d) It measures the standard deviation of the confidence interval when repeated samples are taken from the population.

   (e) It measures how much the standard error would change when a new sample is taken.

   Solution: (c)
   Option B - 16% chose. Population parameters are fixed and do not vary.
   Option C - 70% chose. This doesn't even make sense.

4. (*R*) Which of the following is correct given the following information about a data frame on the composition of cereals:

```
> str(cereal)
'data.frame': 74 obs. of  16 variables:
 $ calories: int  70 120 70 50 110 110 110 130 90 90 ...
 $ protein : int  4 3 4 4 2 2 2 3 2 3 ...
 $ fat     : int  1 5 1 0 2 2 0 2 1 0 ...
 $ shelf   : Factor w/ 3 levels "1","2","3": 3 3 3 3 3 1 2 3 1 3 ...
 $ hot     : num  0 0 0 0 0 0 0 0 0 0 ...
```

   (a) The *lm(fat ~ calories, data=cereal)* is used to test if the mean number of calories differs by the amount of fat in the sample.

   (b) The *glm(hot ~ calories, data=cereal)* function is used to test hypotheses if the mean number of hot cereals (1) or a cold cereals (0) varies by the the number of calories/serving in the sample.

(c) The *lm(shelf ~ calories, data=cereal)* function is used to test if the mean number of calories/serving varies over the different shelves in the population.

(d) The *t.test(calories ~ shelf,data=cereal)* is used to test if the mean number of calories/serving varies over the different shelves in the population.

(e) The *lm(fat ~ protein, data=cereal)* function is used to test hypotheses if the mean number of grams of fat varies by the amount of protein in a serving in the population.

Solution: (e)
Option B - 9% chose.
Option C - 11% chose.
Option D - 34% chose. The *t.test()* function can only be used with 2 levels in a factor. The *shelf* factor has 3 levels.
Option E - 45% chose.

5. (R) Here is some output from the *t.test()* function on the analysis of final grades in a course by the sex of the student.

```
        Welch Two Sample t-test

data:  grade by sex
t = 1.1489, df = 37.421, p-value = 0.2579
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.753133  9.970635
sample estimates:
mean in group f mean in group m
       79.79441        76.18566
```

Which of the following is correct?

(a) There is 26% chance that there is no difference in sex between the grades.

(b) I am about 95% confident that individual grades lie beween -3 and 10.

(c) The p-value of 0.2579 indicates that there is about a 26% chance that there is a difference in mean grade between students.

(d) Because the confidence interval does cover zero, there is no evidence of a difference between the mean grades of the two sexes.

(e) The test statistic ($t = 1.1489$) is the estimated difference in means between the two sexes.

Solution: (d)
Option B - 36% chose. Confidence intervals say NOTHING about individual grades.
Option C - 20% chose. This is the wrong interpretation of a p-value.
Option D - 30% chose.

6. (*R*) Which of the following is a correct statement?

   (a) An *R* vector can contain both integer and logical values.

   (b) An *R* list can contain vectors, arrays, and *glm()* objects.

   (c) An *R* data frame can have different number of rows for each column of data.

   (d) An *R* function can return multiple objects without using a list.

   (e) An *R* matrix must always have the same number of rows and columns.

   Solution: (b)
   Option A - XX% chose.
   Option C - XX% chose. Option E - XX% chose.

7. (*R*) The following section of code was run.

```
sex <- factor(c('small','large','medium'), levels=c('large','medium','small'))
str(sex)
```

   Which of the following is the correct output from the *str()* function?

   (a) Factor w/ 3 levels "large","medium",..: 1 2 3

   (b) Factor w/ 3 levels "large","medium",..: 1 3 2

   (c) Factor w/ 3 levels "large","medium",..: 3 2 1

   (d) Factor w/ 3 levels "large","medium",..: 3 1 2

   (e) Factor w/ 3 levels "large","medium",..: 2 1 3

   Solution: (d)
   Option D - 86% chose.

8. (*R*) Here is some output from the *lm()* function on the analysis of the grades (out of 100) over time for male students in Stat-340.

```
Call:
lm(formula = grade ~ year, data = grade.df, subset = grade.df$sex ==
    "m")

Residuals:
     Min       1Q   Median       3Q      Max
-26.0266  -5.4986   0.1099   6.4180  20.8442

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -678.1071  1442.8373  -0.470    0.639
year           0.3765     0.7160   0.526    0.600
```

```
Residual standard error: 10.13 on 98 degrees of freedom
Multiple R-squared: 0.002813,        Adjusted R-squared: -0.007363
F-statistic: 0.2764 on 1 and 98 DF,  p-value: 0.6002
```

Which of the following is correct?

(a) The estimated regression line is silly because the intercept is negative.

(b) Student grades decline, on average, by about 0.38 marks/year.

(c) Because the *p*-value is 0.600, there is about a 60% chance that the hypothesis of no change in grades over time is true.

(d) The 95% confidence interval for the slope includes the value of 0 and so there is no evidence that the mean grade changes over time.

(e) The standard error for the slope of 0.72 measures how much the grades vary over students in any particular year.

Solution: (d)
Option D - 80% chose.
Option E - 11% chose. Again, SE do NOT measure INDIVIDUAL variation.

9. (*R*) What is the result of the following *R* code?

```
x1<- matrix(c(1,2,3,3,7,8,8,6,1), nrow=3, ncol=3, byrow=FALSE)
apply(x1,2,median)
```

(a) c(2, 7, 6)

(b) c(4, 5, 4)

(c) c(3, 6, 3)

(d) c(2, 6, 5)

(e) c(1, 3, 8)

Solution: (a)
Option A - 64% chose.
Option C - 27% chose. Try it and see the result.

10. (*R*) The following section of code was run.

```
sum(3 + c(1,2,3))
```

Which of the following represents the correct output?

(a) 9

(b) 15

(c) c(4,5,6)

(d) c(1,2,3)

(e) 18

Solution: (b)
Option B - 93% chose.

11. (*SAS*) How many observations and variables are contained in the following dataset?

```
data blah;
    infile datalines;
    length name $10 sex $1;
    input name sex age;
    if sex ="M" then delete;
    datalines;
Carl     M   56
Lois      .   43
 .          .    .
Matthew M   26
Marianne F   23
David    M   22
;;;;
```

(a) 6 observations; 3 variables.

(b) 5 observations, 3 variables.

(c) 3 observation, 6 variables.

(d) 3 observations, 5 variables.

(e) 3 observations, 3 variables.

Solution: (e)
Option A - 51% chose.
Option B - 12% chose.
Option D - 37% chose.

12. (*SAS*) The *MISSOVER* option on the *INFILE* statement performs what function?

(a) Issues an error message and stops *SAS* if a data line has fewer values than variables on an INPUT statement.

(b) Goes to the next record if a date line has fewer values than variables on an INPUT statement and issues a warning message.

(c) Allows an input record to have more data values than variables on the INPUT statement.

(d) Because 256 characters is the default length for *SAS* input records, this option extends the maximum length of an input record.

(e) Inserts missing values into variables that try to read past the last data value on the input record.

Solution: (e)
Option E - 91% chose.

13. (*SAS*) Which of the following is correct?

(a) *PROC GLM* is used to test hypotheses about mean proportions.

(b) *PROC FREQ* is used to test hypotheses about mean proportions.

(c) *PROC REG* is used to test hypotheses about proportional means.

(d) *PROC GENMOD* is used to test hypotheses about population proportions.

(e) *PROC TTEST* is used to test hypotheses about sample means.

Solution: (d)
Option D - 88% chose.
Option E - 12% chose. Hypotheses are ALWAYS about POPULATION parameters, not sample statistics.

14. (*SAS*) Which of the following is INCORRECT about the bootstrap method to determine standard errors as seen in this class?

(a) We compute the estimate for every bootstrap sample.

(b) Bootstrap samples are selected with replacement from the given sample with the same sample size.

(c) The average of the estimates over the bootstrap samples measures the standard error.

(d) About 1000 bootstrap samples should be chosen.

(e) The 95% confidence interval is found using the 2.5^{th} and 97.5^{th} percentile of the bootstrap estimates.

Solution: (c)
Option C - 86% chose.

15. (*SAS*) Which informat is needed to read date values of the form "2013-04-24" (excluding the quotes)?

(a) input mydate:ymd10.;

(b) input mydate:yymmdd10.;

(c) input mydate:yy-mm-dd10.;

(d) input mydate:y-m-d10.;

(e) input mydate:yyyymmdd10.;

Solution: (b) Option B - 49% chose.
Option C - 12% chose.
Option E - 33% chose. There is no informat that stars with 4 Y's.

16. (*SAS*) Here is some output from *PROC REG* on the analysis of the change in grades (out of 100) over time for male students in Stat-340.

Model	Dependent	Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Lower 95% CL Parameter	Upper 95% CL Parameter
MODEL1	grade	Intercept	1	343.23094	1315.50614	0.26	0.7947	-2267.34817	2953.81005
MODEL1	grade	year	1	-0.12942	0.65286	-0.20	0.8433	-1.42499	1.16615

Which of the following is correct?

(a) The estimated regression line has a negative slope indicating individual students are getting worse over time.

(b) The intercept measures the average grade of students in the years studied.

(c) Because the p-value is 0.84, there is an 84% chance that the hypothesis of no changes in individual student grades over time is true.

(d) Because the p-value for the slope is large, there no evidence that the mean grade changes over time.

(e) The standard error for the slope measures how much the grades vary over students in any particular year.

Solution: (d)
Option D - 63% chose.
Option E - 16% chose. Again, SE do NOT measure INDIVIDUAL variation.

17. (*SAS*) Here are two data sets that are merged into a final dataset.

```
data ds1;
   input studentid name$ height;
   datalines;
1         12    Carla    150
2        123     Carl      .
3        456     Fred    190
4        789 Marianne    155
;;;;

data ds2;
   input studentid weight;
   datalines;
1         12     50
2        175     85
3        456     90
4        899     55
;;;;
```

```
data allds;
   merge ds1 ds2; by studentid;
run;
```

Which statement is FALSE?

(a) The first observation will have 12 Carla 150 50 as data values for the four variables.

(b) The second observation will have 123 Carl . 85 as the data values for the four variables.

(c) The third observation will have 175 . . 85 as the data values for the four variables.

(d) The fourth observation will have 456 Fred 190 90 as the data values for the four variables.

(e) The fifth observation will have 789 Marianne 155 . as the data values for the four variables.

Solution: (b)
Option B - 95% chose.

18. (*SAS*) Consider the following code fragment to find the average grade from assignments for each student.

```
data assign;
   input studentid assign mark;
   datalines;
123  1  18
123  2  12
456  1  18
789  2  19
789  3  17
;;;;

proc means data=assign noprint;
  by studentid;
  var mark;
  output out=assign_avg mean=mean_assign;
run;

proc print data=assign_avg;
run;
```

Which statement is correct?

(a) The mean assignment mark for student 123 is 15.

(b) The mean assignment mark for student 456 is 6.

(c) The mean assignment mark for student 789 is 12.

(d) The mean assignment mark for all students is just over 17.

(e) The mean assignment mark cannot be computed because of missing assignments.

Solution: (a)
Option A - 88% chose.

19. (*SAS*) Consider the following code fragment to find the average grade from assignments for each student.

```
data assign;
   input studentid assign1 assign2 assign3;
   avg = (assign1 + assign2 + assign3)/3;
   datalines;
123  18  12 .
456  18  .   .
789  12 19  17
;;;;
```

Which statement is correct?

(a) The mean assignment mark for student 123 is 15.

(b) The mean assignment mark for student 123 is 10.

(c) The mean assignment mark for student 789 is 12.

(d) The mean assignment mark for student 456 is missing.

(e) The mean assignment mark for all students is computed to be 16.

Solution: (d)
Option A - 12% chose. The missing value for assignment 3 implies that the result is missing.
Option B - 42% chose. The missing value for assignment 3 implies that the result is missing.
Option D - 47% chose.

20. (*SAS*) Which statement is correct about *Proc SGplot*?

(a) The *scatter* statement plots the points and then fits a line to the data points.

(b) The *series* statement plots the points and then fits a linear regression to the data points.

(c) The *highlow* statement joins points in a regression line.

(d) The *density* statement creates a histogram of the data value.

(e) The *band* statement draws a shaded band between an upper and lower bound.

Solution: (e)
Option A - 14% chose. The *scatter* statement does not fit a line to the data points.
Option B - 14% chose. The *series* statement joins the individual points with line segments, but does not a regression line.
Option D - 60% chose.

Relationship of question scores to total score on this part

Item-total Statistics

	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Alpha if Item Deleted
Q#1	13.1818	8.4313	.3764	.6269
Q#2	12.7727	9.9937	-.1509	.6819
Q#3	12.9091	8.8753	.2506	.6438
Q#4	13.1591	9.0206	.1663	.6552
Q#5	12.9091	8.5497	.3760	.6281
Q#6	13.2500	9.0291	.1750	.6536
Q#7	12.7500	9.0291	.3010	.6400
Q#8	12.8182	8.8034	.3335	.6349
Q#9	12.9773	8.7204	.2854	.6394
Q#10	12.6818	9.1057	.3943	.6371
Q#11	13.2500	9.2151	.1102	.6617
Q#12	12.7273	9.1797	.2543	.6448
Q#13	12.7500	8.5174	.5624	.6151
Q#14	12.7727	8.8774	.3462	.6349
Q#15	13.1364	9.2368	.0929	.6648
Q#16	13.0000	8.5581	.3389	.6322
Q#17	12.6818	9.0127	.4571	.6328
Q#18	12.7500	9.1686	.2323	.6463
Q#19	13.1591	9.2532	.0883	.6653
Q#20	13.0227	9.0925	.1459	.6576

2 Part II -Long Answer

Name

Student Number:

Put your name and student number on the upper right of each of the following pages as well in case the pages get separated.

Answer the following four questions in the space provided. Stat-341 students should only answer the first 2 question (on using *R*). Be sure that your answers are legible.

The marks given to these four questions are 4, 6, 6 and 4 respectively.

1. Non-parametric estimate of slope in simple linear regression - using *R*

A non-parametric estimate of the slope in simple linear regression finds the slopes for all possible pairs of points and then finds the median of these value, i.e.

$$\widehat{\beta} = \underset{i<j}{median} \frac{Y_i - Y_j}{X_i - X_j}$$

where Y_i and X_i are the components of the i^{th} point.

Construct an *R* function (named *find.slope* that takes as input two arguments (Y and X) and returns a two element vector which contains the slope from a ordinary least squares regression of Y on X (named *reg.slope*) and the non-parametric estimate of the slope (named *np.slope*).

You may assume that there are NO missing values in either Y or X.

Hint: Construct all the possible $Y_i - Y_j$ using the *outer(Y,Y,FUN='-')* and construct the selection matrix to select the points such that $i < j$ using the *outer(1:n,1:n,FUN="<")*.

One possible solution

```
find.slope <- function(X,Y){
# Find the regular and non-parametric estimate of the slope

# Regular slope
fit.lm <- lm(Y ~ X)
reg.slope <- coef(fit.lm)[2]

# Non-parametric slope
n <- length(Y)
Y.diff <- outer(Y,Y,FUN='-')
X.diff <- outer(X,X,FUN='-')
Sel    <- outer(1:n,1:n,FUN='<')
np.slope <- median( (Y.diff/X.diff)[Sel])

return(c(reg.slope=reg.slope, np.slope=np.slope))
}
```

```
> set.seed(12342)
> n <- 5
> reg   <- NULL
> reg$X <- runif(n,min=10, max=30)
> reg$Y <- 3 + 2*reg$X + rnorm(n)*5
>
> find.slope(reg$X, reg$Y)
reg.slope.X    np.slope
   1.408009    1.861192
>
```

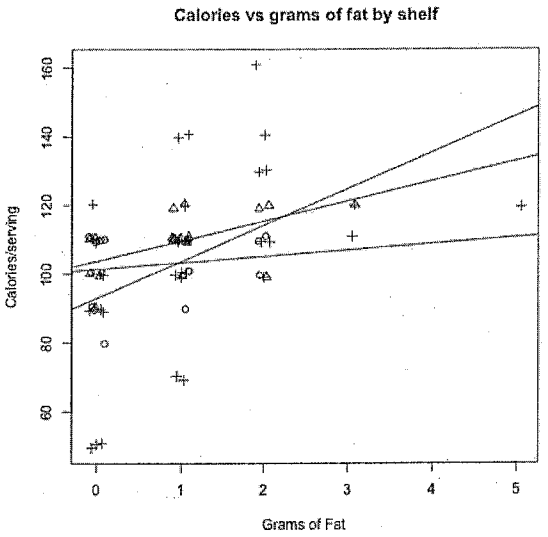
2. BY group processing in *R*

Write a series of *R* statements to do the following:

- Read in the cereal data set from the cereal.csv file. You may assume that all the variable names are in the first row.
- Print the first 10 records of the data frame.
- The data frame contains a (numeric) variable *shelf* that contains the shelf number (1, 2, 3, or 9) where 9 indicates that the shelf is unknown. REMOVE all the lines in the data frame where the shelf is unknown.
- For each shelf, compute the regression of calories/serving (*calories*) against grams of fat (*fat*). The resulting object should contain all of the results in a list.
- Plot the calories/serving vs the grams of fat using a different symbol for each shelf and jittering the points to prevent over plotting. A jitter of about .1 for the fat value and 1 for the calories values would suffice. Be sure to label the axes and give an appropriate title.
- Extract the slope and intercept for the regression for each shelf and store this in a matrix.
- Plot the fitted lines on the earlier plot – one line for each shelf.
- Send the plot to an external png file.

Solution:

```
cereal <- read.csv('cereal.csv', header=TRUE)
cereal[1:10,]

# Remove all rows corresponding to shelf 9 = missing
cereal <- cereal[ !cereal$shelf==9,]

# calories vs grams of fat for each shelf
fit.lm <- with(cereal, by(cereal, shelf, function(x){
   lm(calories ~ fat, data=x)
}))

# plot of the calories/serving vs the grams of fat using a different symbol for
# Jitter the fat and calories values

# Extract the intercept and slope from the regression of each slope
slopes <- sapply(fit.lm, coef) # notice interecept and slopes are stored in colu
slopes

png('final-cal-vs-fat.png')
with(cereal, plot(fat+runif(length(fat),min=-.1,max=.1),
                  calories+runif(length(calories),min=-1,max=1), pch=shelf,
                  main='Calories vs grams of fat by shelf',
                  xlab='Grams of Fat',ylab='Calories/serving'))
apply(slopes,2,function(x){abline(x)})
dev.off()
```



Many students set the shelf variable to NA for the unknown shelf values, but did not REMOVE the line from the data frame. Many students added small values to ALL the co-ordinates. This will not jitter the points as all points are affected equally.

3. Computing Final Grades - SAS

Write *SAS* code to do the following:

- Read from a csv file (file name is *grades.csv*) data with a separate data record for each student containing
  - student number - 8 digits character;
  - grade on test 1 (out of 20) - numeric
  - grade on test 2 (out of 20) - numeric
  - grade on the final (out of 40) - numeric
  - average grade on assignments (out of 20) - numeric.

  Here is some sample data:

```
12345678 , 15 , 16, 35,  17
19234924 , 13 , 17, 37,  19
48304393 , 10 , 10, 15,   5
23408420 , 12 , 12, 24,  12
23423443 ,  . , 15, 35,  18
```

- Print the first 10 records of the read dataset.
- Compute, for each student, their overall grade (out of 100) as the sum of the 4 grades listed above. However, some students missed term test 1 (no students missed term test 2 or the final). For these students, compute the overall grade (out of 100) as the sum of grade on test 2, the grade on the final prorated to be out of 60, and the grade on on the assignment.

- Create a report for posting the overall grades on an office door. To preserve anonymity, you must extract the last 4 digits of the student number, sort by these last 4 digits, and print a report that only shows the last 4 digits of the student number and the overall grade.
- Computes the average and standard deviation of the overall grade.

Solution:

```
title 'Computation of overall grades ';
data grades;
   infile datalines dlm=',' dsd missover;
   length snumber $8;
   input snumber test1 test2 final assign;
   datalines;
12345678 , 15 , 16, 35,  17
19234924 , 13 , 17, 37,  19
48304393 , 10 , 10, 15,   5
23408420 , 12 , 12, 24,  12
23423443 ,  . , 15, 35,  18
;;;;

proc print data=grades(obs=10);
   title2 'partial listing of grades';
run;

/* Compute the overall and letter grade */
data grades;
   set grades;
   overall = test1 + test2 + final + assign;
   if test1 = . then do;
      overall = test2 + final/40*60 + assign;
   end;
run;

/* make the simple report */
data simple;
   set grades;
   snumber_last4 = substr(snumber,5,4);
   keep snumber_last4 overall;
run;
proc sort data=simple; by snumber_last4; run;
proc print data=simple;
   title2 'Final letter grades sorted by last 4 digits of student number';
run;

/* find the overall average */
proc univariate data=grades;
   title2 'Overall average and std deviation';
```

```
   var overall;
run;
```

4. BY group processing in *SAS*

The *SAS* dataset (*cereal*) has been read containing information on composition of cereals. Write a series of SAS statements to do the following:

- The data set contains a (numeric) variable *shelf* that contains the shelf number (1, 2, 3, or 9) where 9 indicates that the shelf is unknown. Remove all the records from the data where the shelf number is unknown.
- For each shelf, compute the regression of calories/serving (*calories*) against grams of fat (*fat*).
- Send the estimates of the slope and intercept for each shelf group to a data set using the ODS feature. The ODS table name is *ParameterEstimates*.
- Print the data set containing the estimates, but only display the estimates and their standard error to 1 decimal place. Change the label for the standard error variable to SE. The variable names containing the estimates and standard errors are *Estimate* and *StdErr* respectively.

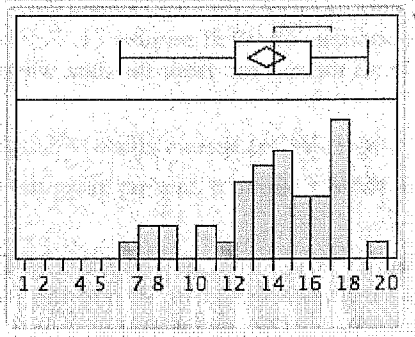
Solution:

```
/* delete all records with shelf = 9 */
data cereal;
   set cereal;
   if shelf = 9 then delete;
run;

/* Run separate regressions */
proc sort data=cereal; by shelf; run;

proc reg data=cereal;
   title2 'regression of calories by fat for each shelf';
   by shelf;
   model calories = fat;
   ods output ParameterEstimates= Estimates;
run;

proc print data=Estimates label split=' ';
   title2 'Extracted estimates';
   format Estimate StdErr 7.1;
   label StdErr='SE';
run;
```

Statistics about the final exam:

Distributions

MC



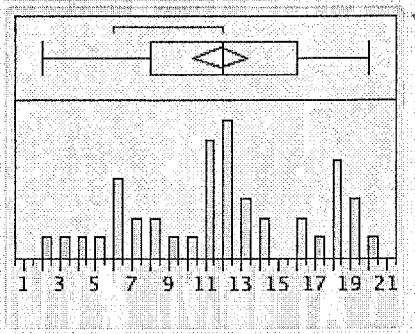
Quantiles

100.0%	maximum	19
99.5%		19
97.5%		18.75
90.0%		17
75.0%	quartile	16
50.0%	median	14
25.0%	quartile	12
10.0%		8
2.5%		6.125
0.5%		6
0.0%	minimum	6

Summary Statistics

Mean	13.613636
Std Dev	3.1268969
Std Err Mean	0.4713974
Upper 95% Mean	14.5643
Lower 95% Mean	12.662973
N	44

Long Answer



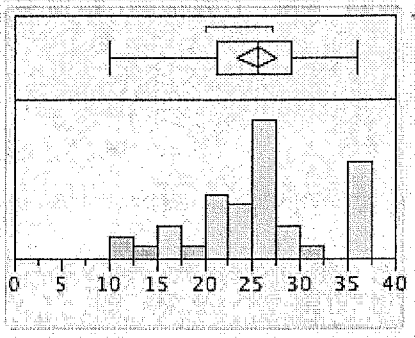
Quantiles

100.0%	maximum	20
99.5%		20
97.5%		19.875
90.0%		18.5
75.0%	quartile	16
50.0%	median	12
25.0%	quartile	8
10.0%		5.5
2.5%		2.125
0.5%		2
0.0%	minimum	2

Summary Statistics

Mean	11.818182
Std Dev	4.8140914
Std Err Mean	0.7257516
Upper 95% Mean	13.281799
Lower 95% Mean	10.354564
N	44

Total



Quantiles

100.0%	maximum	36
99.5%		36
97.5%		36
90.0%		35
75.0%	quartile	29
50.0%	median	25.5
25.0%	quartile	21.25
10.0%		16
2.5%		10.125
0.5%		10
0.0%	minimum	10

Summary Statistics

Mean	25.431818
Std Dev	6.7768338
Std Err Mean	1.0216461
Upper 95% Mean	27.492164
Lower 95% Mean	23.371472
N	44

Operators

<-	Left assignment, binary
->	Right assignment, binary
=	Left assignment, but not recommended
<<-	Left assignment in outer lexical scope; not for beginners
$	List subset, binary
-	Minus, can be unary or binary
+	Plus, can be unary or binary
~	Tilde, used for model formulae
:	Sequence, binary (in model formulae: interaction)
::	Refer to function in a package, i.e, pkg::function; usually not needed
*	Multiplication, binary
/	Division, binary
^	Exponentiation, binary
%x%	Special binary operators, x can be replaced by any valid name
%%	Modulus, binary
%/%	Integer divide, binary
%*%	Matrix product, binary
%o%	Outer product, binary
%x%	Kronecker product, binary
%in%	Matching operator, binary (in model formulae: nesting)
! x	logical negation, NOT x
x & y	elementwise logical AND
x && y	vector logical AND
x | y	elementwise logical OR
x || y	vector logical OR
xor(x, y)	elementwise exclusive OR
<	Less than, binary
>	Greater than, binary
==	Equal to, binary
>=	Greater than or equal to, binary
<=	Less than or equal to, binary

Packages

install.packages("pkgs", lib) download and install pkgs from repository (lib) or other external source

update.packages checks for new versions and offers to install

library(pkg) loads pkg, if pkg is omitted it lists packages

detach("package:pkg") removes pkg from memory

Indexing vectors

x[n]	nth element
x[-n]	all but the nth element
x[1:n]	first n elements
x[-(1:n)]	elements from n+1 to end
x[c(1,4,2)]	specific elements
x["name"]	element named "name"
x[x > 3]	all elements greater than 3
x[x > 3 & x < 5]	all elements between 3 and 5
x[x %in% c("a","if")]	elements in the given set

Indexing lists

x[n]	list with elements n
x[[n]]	nth element of the list
x[["name"]]	element named "name"
x$name	as above (w. partial matching)

Indexing matrices

x[i,j]	element at row i, column j
x[i,]	row i
x[,j]	column j
x[,c(1,3)]	columns 1 and 3
x["name",]	row named "name"

Indexing matrices data frames (same as matrices plus the following)

X[["name"]]	column named "name"
x$name	as above (w. partial matching)

Input and output (I/O)

R data object I/O

data(x) loads specified data set; if no arg is given it lists all available data sets

save(file,...) saves the specified objects (...) in XDR platform-independent binary format

save.image(file) saves all objects

load(file) load datasets written with save

Database I/O

Useful packages: *DBI* interface between R and relational DBMS; *RJDBC* access to databases through the JDBC interface; *RMySQL* interface to MySQL database; *RODBC* ODBC database access; *ROracle* Oracle database interface driver; *RpgSQL* interface to PostgreSQL database; *RSQLite* SQLite interface for R

wf("file") read a
or a
lth file and create a

,file) saves x after

nscript of
t text file

et

e arguments after
the character
;
generic, meaning it
or different objects
for pretty printing
k()

also be used to read
of a file.
ipe("pbpaste"))
"clipboard")

arguments with the
h recursive=TRUE
ning all elements

" has operator

ce by= specifies
desired length
length(along);

se each to repeat
es; rep(c(1,2,3),2) is
h=2) is 1 1 2 2 3 3
ne of the named or
ne (v=1:4, ch=
orter vectors are
longest
or unnamed
ii", c=3);

array(x,dim=) array with data x; specify dimensions like dim=c(3,4,2); elements of x recycle if x is not long enough
matrix(x,nrow,ncol) matrix; elements of x recycle
factor(x,levels) encodes a vector x as a factor
gl(n, k, length=n*k, labels=1:n) generate levels (factors) by specifying the pattern of their levels; k is the number of levels, and n is the number of replications
expand.grid() a data frame from all combinations of the supplied vectors or factors

Data conversion

as.array(x), as.character(x), as.data.frame(x), as.factor(x), as.logical(x), as.numeric(x), convert type; for a complete list, use **methods(as)**

Data information

is.na(x), is.null(x), is.nan(x); is.array(x), is.data.frame(x), is.numeric(x), is.complex(x), is.character(x); for a complete list, use **methods(is)**
x prints x
head(x), tail(x) returns first or last parts of an object
summary(x) generic function to give a summary
str(x) display internal structure of the data
length(x) number of elements in x
dim(x) Retrieve or set the dimension of an object; **dim(x) <- c(3,2)**
dimnames(x) Retrieve or set the dimension names of an object
nrow(x), ncol(x) number of rows/cols; **NROW(x), NCOL(x)** is the same but treats a vector as a one-row/col matrix
class(x) get or set the class of x; **class(x) <- "myclass";**
unclass(x) removes the class attribute of x
attr(x,which) get or set the attribute which of x
attributes(obj) get or set the list of attributes of obj

Data selection and manipulation

which.max(x), which.min(x) returns the index of the greatest/smallest element of x
rev(x) reverses the elements of x
sort(x) sorts the elements of x in increasing order; to sort in decreasing order: **rev(sort(x))**
cut(x,breaks) divides x into intervals (factors); breaks is the number of cut intervals or a vector of cut points
match(x, y) returns a vector of the same length as x with the elements of x that are in y (NA otherwise)
which(x == a) returns a vector of the indices of x if the comparison operation is true (TRUE), in this example the values of i for which x[i] == a (the argument of this function must be a variable of mode logical)
choose(n, k) computes the combinations of k events among n repetitions = n!/[(n − k)!k!]
na.omit(x) suppresses the observations with missing data (NA)
na.fail(x) returns an error message if x contains at least one NA
complete.cases(x) returns only observations (rows) with no NA
unique(x) if x is a vector or a data frame, returns a similar object but with the duplicates suppressed
table(x) returns a table with the numbers of the different values of x (typically for integers or factors)
split(x, f) divides vector x into the groups based on f
subset(x, ...) returns a selection of x with respect to criteria (..., typically comparisons: x$V1 < 10); if x is a data frame, the option select gives variables to be kept (or dropped, using a minus)
sample(x, size) resample randomly and without replacement size elements in the vector x, for sample with replacement use: replace = TRUE
sweep(x, margin, stats) transforms an array by sweeping out a summary statistic
prop.table(x,margin) table entries as fraction of marginal table
xtabs(a b,data=x) a contingency table from cross-classifying factors
replace(x, list, values) replace elements of x listed in index with values

es by common col

b[separate cols
ing.. col

upplied matrices,
ols
rs) changes an
easy casting,

fun to melted data
age)
casts in a single

data frame
surements in
ated measurements
l on direction

or, d=dataframe)
put: a or l; applies
(index) of x
ply fun to each

ser friendly
replicate()
ıt l; applies fun to
on index
ut is class "by",

if; applies fun
ed on index. Can

nean (or other fun)
by

onsistent names:
e, second is
l(ist), a(rray), or
hree main

fun, ...)
.fun, ...)

ith ply functions
transform()

Math

Many math functions have a logical parameter na.rm=FALSE to specify missing data removal.

sin,cos,tan,asin,acos,atan,atan2,log,log10,exp
min(x), max(x) min/max of elements of x
range(x) min and max elements of x
sum(x) sum of elements of x
diff(x) lagged and iterated differences of vector x
prod(x) product of the elements of x
round(x, n) rounds the elements of x to n decimals
log(x, base) computes the logarithm of x
scale(x) centers and reduces the data; can center only (scale=FALSE) or reduce only (center=FALSE)
pmin(x,y,...), pmax(x,y,...) parallel minimum/maximum, returns a vector in which ith element is the min/max of x[i], y[i], . . .
cumsum(x), cummin(x), cummax(x), cumprod(x) a vector which ith element is the sum/min/max from x[1] to x[i]
union(x,y), intersect(x,y), setdiff(x,y), setequal(x,y), is.element(el,set) "set" functions
Re(x) real part of a complex number
Im(x) imaginary part
Mod(x) modulus; **abs(x)** is the same
Arg(x) angle in radians of the complex number
Conj(x) complex conjugate
convolve(x,y) compute convolutions of sequences
fft(x) Fast Fourier Transform of an array
mvfft(x) FFT of each column of a matrix
filter(x,filter) applies linear filtering to a univariate time series or to each series separately of a multivariate time series

Correlation and variance

cor(x) correlation matrix of x if it is a matrix or a data frame (1 if x is a vector)
cor(x, y) linear correlation (or correlation matrix) between x and y
var(x) or **cov(x)** variance of the elements of x (calculated on n − 1); if x is a matrix or a data frame, the variance-covariance matrix is calculated
var(x, y) or **cov(x, y)** covariance between x and y, or between the columns of x and those of y if they are matrices or data frames

Matrices

t(x) transpose
diag(x) diagonal
%*% matrix multiplication
solve(a,b) solves a %*% x = b for x solve(a) matrix inverse of a
rowsum(x), colsum(x) sum of rows/cols for a matrix-like object (consider **rowMeans(x), colMeans(x)**)

Distributions

Family of distribution functions, depending on first letter either provide: r(andom sample) ; p(robability density), c(umulative probability density),or q(uantile):

rnorm(n, mean=0, sd=1) Gaussian (normal)
rexp(n, rate=1) exponential
rgamma(n, shape, scale=1) gamma
rpois(n, lambda) Poisson
rweibull(n, shape, scale=1) Weibull
rcauchy(n, location=0, scale=1) Cauchy
rbeta(n, shape1, shape2) beta
rt(n, df) 'Student' (t)
rf(n, df1, df2) Fisher-Snedecor (*F*) (!!!²)
rchisq(n, df) Pearson
rbinom(n, size, prob) binomial
rgeom(n, prob) geometric
rhyper(nn, m, n, k) hypergeometric
rlogis(n, location=0, scale=1) logistic
rlnorm(n, meanlog=0, sdlog=1) lognormal
rnbinom(n, size, prob) negative binomial
runif(n, min=0, max=1) uniform
rwilcox(nn, m, n), rsignrank(nn, n) Wilcoxon

Descriptive statistics

mean(x) mean of the elements of x
median(x) median of the elements of x
quantile(x,probs=) sample quantiles corresponding to the given probabilities (defaults to 0,.25,.5,.75,1)
weighted.mean(x, w) mean of x with weights w
rank(x) ranks of the elements of x
describe(x) statistical description of data (in *Hmisc* package)
describe(x) statistical description of data useful for psychometrics (in *psych* package)
sd(x) standard deviation of x
density(x) kernel density estimates of x

df.residual(fit) returns residual degrees of freedom
coef(fit) returns the estimated coefficients (sometimes with standard-errors)
residuals(fit) returns the residuals
deviance(fit) returns the deviance
fitted(fit) returns the fitted values
logLik(fit) computes the logarithm of the likelihood and the number of parameters
AIC(fit), BIC(fit) compute Akaike or Bayesian information criterion
influence.measures(fit) diagnostics for lm & glm
approx(x,y) linearly interpolate given data points; x can be an xy plotting structure
spline(x,y) cubic spline interpolation
loess(formula) fit polynomial surface using local fitting
optim(par, fn, method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN") general-purpose optimization; par is initial values, fn is function to optimize (normally minimize)
nlm(f,p) minimize function f using a Newton-type algorithm with starting values p

Flow control

if(cond) expr
if(cond) cons.expr else alt.expr
for(var in seq) expr
while(cond) expr repeat expr
break
next
switch
Use braces {} around statements
ifelse(test, yes, no) a value with the same shape as test filled with elements from either yes or no
do.call(funname, args) executes a function call from the name of the function and a list of arguments to be passed to it

Writing functions

function(arglist) expr function definition,
missing test whether a value was specified as an argument to a function
require load a package within a function
<<- attempts assignment within parent environment before search up thru environments
on.exit(expr) executes an expression at function end
return(value) or **invisible**

Strings

paste(vectors, sep, collapse) concatenate vectors after converting to character; sep is a string to separate terms; collapse is optional string to separate "collapsed" results; see also str_c below
substr(x,start,stop) get or assign substrings in a character vector. See also str_sub below
strsplit(x,split) split x according to the substring split
grep(pattern,x) searches for matches to pattern within x; see **?regex**
gsub(pattern,replacement,x) replace pattern in x using regular expression matching; sub() is similar but only replaces the first occurrence.
tolower(x), toupper(x) convert to lower/uppercase
match(x,table) a vector of the positions of first matches for the elements of x among table
x %in% table as above but returns a logical vector
pmatch(x,table) partial matches for the elements of x among table
nchar(x) # of characters. See also str_length below

***stringr* package** provides a nice interface for string functions:
str_detect detects the presence of a pattern; returns a logical vector
str_locate locates the first position of a pattern; returns a numeric matrix with col start and end. (str_locate_all locates all matches)
str_extract extracts text corresponding to the first match; returns a character vector (str_extract_all extracts all matches)
str_match extracts "capture groups" formed by () from the first match; returns a character matrix with one column for the complete match and one column for each group
str_match_all extracts "capture groups" from all matches ; returns a list of character matrices
str_replace replaces the first matched pattern; returns a character vector
str_replace_all replaces all matches.
str_split_fixed splits string into a fixed number of pieces based on a pattern; returns character matrix
str_split splits a string into a variable number of pieces; returns a list of character vectors
str_c joins multiple strings, similar to paste
str_length gets length of a string, similar to nchar
str_sub extracts substrings from character vector, similar to substr

Graphs

There are three main classes of plots in R: base plots, grid & lattice plots, and *ggplot2* package. They have limited interoperability. Base, grid, and lattice are covered here. *ggplot2* needs its own reference sheet.

Base graphics

Common arguments for base plots:

add=FALSE if TRUE superposes the plot on the previous one (if it exists)

axes=TRUE if FALSE does not draw the axes and the box

type="p" specifies the type of plot, "p": points, "l": lines, "b": points connected by lines, "o": same as previous but lines are over the points, "h": vertical lines, "s": steps, data are represented by the top of the vertical lines, "S": same as previous but data are represented by the bottom of the vertical lines

xlim=, ylim= specifies the lower and upper limits of the axes, for example with xlim=c(1, 10) or xlim=range(x)

xlab=, ylab= annotates the axes, must be variables of mode character main= main title, must be a variable of mode character

sub= sub-title (written in a smaller font)

Base plot functions

plot(x) plot of the values of x (on the y-axis) ordered on the x-axis

plot(x, y) bivariate plot of x (on the x-axis) and y (on the y-axis)

hist(x) histogram of the frequencies of x

barplot(x) histogram of the values of x; use horiz=TRUE for horizontal bars

dotchart(x) if x is a data frame, plots a Cleveland dot plot (stacked plots line-by-line and column-by-column)

boxplot(x) "box-and-whiskers" plot

stripplot(x) plot of the values of x on a line (an alternative to boxplot() for small sample sizes)

coplot(x~y | z) bivariate plot of x and y for each value or interval of values of z

interaction.plot (f1, f2, y) if f1 and f2 are factors, plots the means of y (on the y-axis) with respect to the values of f1 (on the x-axis) and of f2 (different curves); the option fun allows to choose the summary statistic of y (by default fun=mean)

matplot(x,y) bivariate plot of the first column of x vs. the first one of y, the second one of x vs. the second one of y, etc.

fourfoldplot(x) visualizes, with quarters of circles, the association between two dichotomous variables for different populations (x must be an array with dim=c(2, 2, k), or a matrix with dim=c(2, 2) if k=1)

assocplot(x) Cohen-Friendly graph showing the deviations from independence of rows and columns in a two dimensional contingency table

mosaicplot(x) 'mosaic' graph of the residuals from a log-linear regression of a contingency table

pairs(x) if x is a matrix or a data frame, draws all possible bivariate plots between the columns of x

plot.ts(x) if x is an object of class "ts", plot of x with respect to time, x may be multivariate but the series must have the same frequency and dates

ts.plot(x) same as above but if x is multivariate the series may have different dates and must have the same frequency

qqnorm(x) quantiles of x with respect to the values expected under a normal distribution

qqplot(x, y) diagnostic plotr of quantiles of y vs. quantiles of x; see also qqPlot in *cars* package and distplot in *vcd* package

contour(x, y, z) contour plot (data are interpolated to draw the curves), x and y must be vectors and z must be a matrix so that dim(z)= c(length(x), length(y)) (x and y may be omitted). See also filled.contour, image, and persp

symbols(x, y, ...) draws, at the coordinates given by x and y, symbols (circles, squares, rectangles, stars, thermometers or "boxplots") with sizes, colours . . . are specified by supplementary arguments

termplot(mod.obj) plot of the (partial) effects of a regression model (mod.obj)

colorRampPalette creates a color palette (use: colfunc <- colorRampPalette(c("black", "white")); colfunc(10)

Low-level base plot arguments

points(x, y) adds points (the option type= can be used)

lines(x, y) same as above but with lines

text(x, y, labels, ...) adds text given by labels at

se is: plot(x, y,

ıdds text given by
›y side (see axis()
ɔ from the plotting
) draws lines from
'1)
0, code=2) same as
x0,y0) if code=2, at
›oth if code=3; angle
haft of the arrow to

: b and intercept a
:al line at ordinate y
line at abcissa x
sion line given by

ıngle with left, right,
x2, y1, and y2,

inking the points
and y
:gend at the point
by legend
a sub-title
he bottom (side=1),
or on the right (4);
ssa (or ordinates)

axis as small

the coordinates (x,
ı times on the plot
ymbols (type="p")
:ct to optional
efault nothing is

r(...); many can be
:ommands.
eft-justified, 0.5

kground (ex. :
ist of the 657
l with colors())
'n around the plot,
'7", "c", "u" ou "]"

(the box looks like the corresponding character); if bty="n" the box is not drawn

cex a value controlling the size of texts and symbols with respect to the default; the following parameters have the same control for numbers on the axes, cex.axis, the axis labels, cex.lab, the title, cex.main, and the sub-title, cex.sub

col controls the color of symbols and lines; use color names: "red", "blue" see colors() or as "#RRGGBB"; see rgb(), hsv(), gray(), and rainbow(); as for cex there are: col.axis, col.lab, col.main, col.sub

font an integer that controls the style of text (1: normal, 2: italics, 3: bold, 4: bold italics); as for cex there are: font.axis, font.lab, font.main, font.sub

las an integer that controls the orientation of the axis labels (0: parallel to the axes, 1: horizontal, 2: perpendicular to the axes, 3: vertical)

lty controls the type of lines, can be an integer or string (1: "solid", 2: "dashed", 3: "dotted", 4: "dotdash", 5: "longdash", 6: "twodash", or a string of up to eight characters (between "0" and "9") that specifies alternatively the length, in points or pixels, of the drawn elements and the blanks, for example lty="44" will have the same effect than lty=2

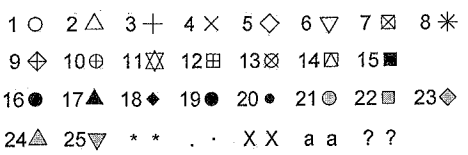
lwd numeric that controls the width of lines, default 1

mar a vector of 4 numeric values that control the space between the axes and the border of the graph of the form c(bottom, left, top, right), the default values are c(5.1, 4.1, 4.1, 2.1)

mfcol a vector of the form c(nr,nc) that partitions the graphic window as a matrix of nr lines and nc columns, the plots are then drawn in columns

mfrow same as above but the plots are drawn by row

pch controls the type of symbol, either an integer between 1 and 25, or any single char within ""



ps an integer that controls the size in points of texts and symbols

pty a character that specifies the type of the plotting region, "s": square, "m": maximal

tck a value that specifies the length of tick-marks on the axes as a fraction of the smallest of the width or height of the plot; if tck=1 a grid is drawn

tcl a value that specifies the length of tick-marks on the axes as a fraction of the height of a line of text (by default tcl=-0.5)

xaxt if xaxt="n" the x-axis is set but not drawn (useful in conjonction with axis(side=1, ...))

yaxt if yaxt="n" the y-axis is set but not drawn (useful in conjonction with axis(side=2, ...))

Lattice graphics

Lattice functions return objects of class trellis and must be printed. Use print(xyplot(...)) inside functions where automatic printing doesn't work. Use lattice.theme and lset to change Lattice defaults. In the normal Lattice formula, y x|g1*g2 has combinations of optional conditioning variables g1 and g2 plotted on separate panels. Lattice functions take many of the same args as base graphics plus also data= the data frame for the formula variables and subset= for subsetting. Use panel= to define a custom panel function (see apropos("panel") and ?llines).

xyplot(y~x) bivariate plots (with many functionalities)

barchart(y~x) histogram of the values of y with respect to those of x

dotplot(y~x) Cleveland dot plot (stacked plots line-by-line and column-by-column)

densityplot(~x) density functions plot histogram(~x) histogram of the frequencies of x bwplot(y~x) "box-and-whiskers" plot

qqmath(~x) quantiles of x with respect to the values expected under a theoretical distribution

stripplot(y~x) single dimension plot, x must be numeric, y may be a factor

qq(y~x) quantiles to compare two distributions, x must be numeric, y may be numeric, character, or factor but must have two 'levels'

splom(~x) matrix of bivariate plots

parallel(~x) parallel coordinates plot

levelplot(z~x*y|g1*g2) coloured plot of the values of z at the coordinates given by x and y (x, y and z are all of the same length)

wireframe(z~x*y|g1*g2) 3d surface plot

cloud(z~x*y|g1*g2) 3d scatter plot