STA 610/615 Fall 2018 Exam 2 Name____________________

Directions: Write clearly and in complete sentences. You may use your book, computer and two sheets of notes. When done, save your file to PDF and upload to the Moodle exam assignment. If you Also turn in your hand written work, if any, and note sheets. Make sure any handwritten work has your name on it.

For each block of R code state what's being done and why:
i) state what the code does,
ii) why it is (or is not) needed and
iii) interpret any output from it.
Be sure to cover everything the R code does in the block. Don't assume that a statement is needed or that it's even correct.

There are several parts to each of the two problems below. Each part is worth about 5 points.

1) The aim of a study by Hargman-Maeir et al. was to evaluate the awareness of deficit profiles among stroke patients undergoing rehabilitation. She studied 35 patients with a stroke lesion in the right hemisphere and 19 patients with a lesion on the left hemisphere. She also grouped lesion size as 2="1-3cm", 3="3-5cm" and 4="4cm or greater". One of the outcome variables was a measure of each patient's total unawareness of their own limitations. Awareness scores ranged from 8-24, with higher scores indicating more unawareness.

2) A study by Kwast-Rabben et.al. analyzed somatosensory evoked potentials (SEPs) and their interrelations following stimulation of digits I, III and V in the hand. Healthy volunteers were recruited for the study. Researchers applied stimulation below-pain-level intensity to the fingers. Recordings of spinal responses were made with electrodes fixed by adhesive electrode cream to the subject's skin. Results (CvDig1) for 114 subjects are analyzed in the output.

STA 610/615 Exam 2 Output

Problem 1

a)

	SCORES <int>	SIDE <fctr>	SIZE <fctr>
	11	L	1-3cm
	13	L	1-3cm
	10	L	1-3cm

1-3 of 54 rows Previous **1** 2 3 4 5 6 ... 18 Next

b)

```
##
## Call:
## lm(formula = SCORES ~ SIDE * SIZE, data = p1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.417  -2.083  -0.200   0.222 164.583
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             10.11111    8.28836   1.220    0.228
## SIDER                   -0.18254   10.62353  -0.017    0.986
## SIZE3-5cm                0.08889   13.86908   0.006    0.995
## SIZE5cm or greater       2.88889   13.86908   0.208    0.836
## SIDER:SIZE3-5cm          0.76032   17.47028   0.044    0.965
## SIDER:SIZE5cm or greater 12.59921   16.97164   0.742    0.461
##
## Residual standard error: 24.87 on 48 degrees of freedom
## Multiple R-squared:  0.06614,    Adjusted R-squared:  -0.03114
## F-statistic: 0.6799 on 5 and 48 DF,  p-value: 0.6408
```

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
SIDE	1	256.3396	256.3396	0.4146061	0.5227058
SIZE	2	1463.3301	731.6650	1.1834018	0.3150100
SIDE:SIZE	2	382.0554	191.0277	0.3089700	0.7356520
Residuals	48	29677.0897	618.2727	NA	NA

4 rows

c)

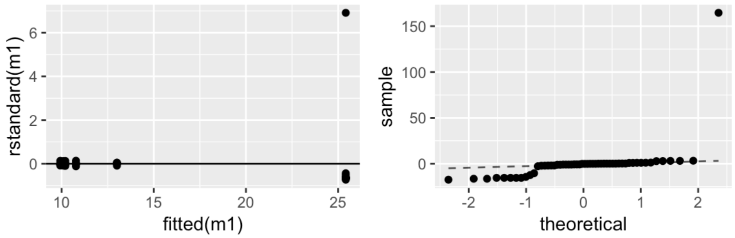
	Df <int>	F value <dbl>	Pr(>F) <dbl>
group	5	3.025978	0.01874698
	48	NA	NA

2 rows

	Df <int>	F value <dbl>	Pr(>F) <dbl>
group	5	0.6917189	0.6321509
	48	NA	NA

2 rows

d)



	SCORES <int>	SIDE <fctr>	SIZE <fctr>
	190	R	5cm or greater

1 row

e)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
SIDE	1	1.287246	1.287246	0.3187572	0.57498453
SIZE	2	29.981129	14.990564	3.7120732	0.03169665
SIDE:SIZE	2	10.540091	5.270046	1.3050072	0.28061742
Residuals	48	193.839683	4.038327	NA	NA

4 rows

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
SIDE	1	1.287246	1.287246	0.3149152	0.57718611
SIZE	2	29.981129	14.990564	3.6673307	0.03264516
Residuals	50	204.379774	4.087595	NA	NA

3 rows

f)

```
##
## Call:
## lm(formula = SCORES ~ SIZE, data = p1a)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7059 -1.0000 -0.2857  1.0000  7.2941
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        10.0000     0.4200  23.807   <2e-16 ***
## SIZE3-5cm           0.5714     0.6829   0.837   0.4066
## SIZE5cm or greater   1.7059     0.6443   2.648   0.0108 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.014 on 51 degrees of freedom
## Multiple R-squared:  0.1218, Adjusted R-squared:  0.08731
## F-statistic: 3.535 on 2 and 51 DF,  p-value: 0.0365
```

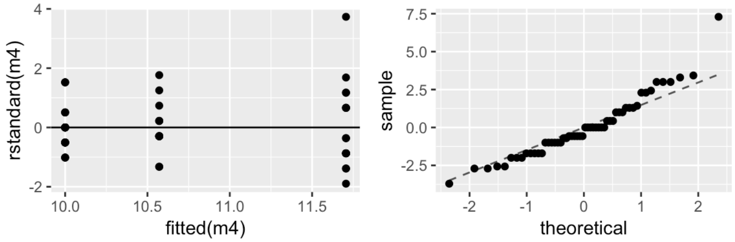
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
SIZE	2	28.69016	14.34508	3.535013	0.03649748
Residuals	51	206.95798	4.05800	NA	NA

2 rows

g)

	Df	F value	Pr(>F)
	<int>	<dbl>	<dbl>
group	2	3.199454	0.04909042
	51	NA	NA

2 rows



h)

```
##
##  Kruskal-Wallis rank sum test
##
## data:  SCORES by SIDE
## Kruskal-Wallis chi-squared = 0.79864, df = 1, p-value = 0.3715
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  SCORES by SIZE
## Kruskal-Wallis chi-squared = 5.0733, df = 2, p-value = 0.07913
```

i)

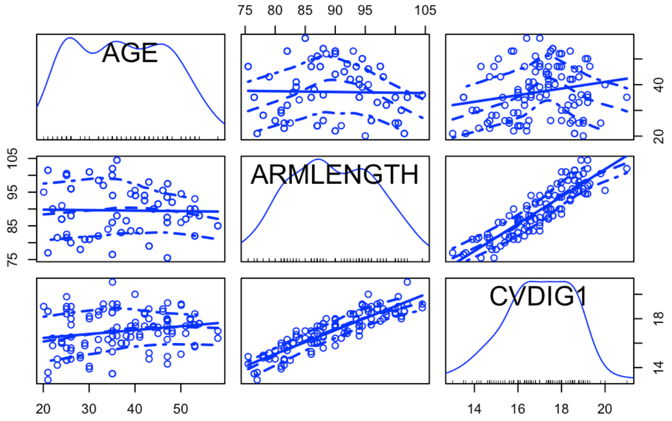
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  SCORES and SIZE
##
##                 1-3cm 3-5cm
## 3-5cm           1.000 -
## 5cm or greater 0.032 0.375
##
## P value adjustment method: bonferroni
```

```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  SCORES and SIZE
##
##                 1-3cm 3-5cm
## 3-5cm           0.602 -
## 5cm or greater 0.099 1.000
##
## P value adjustment method: bonferroni
```

Problem 2

a)



b)

```
##                 AGE    ARMLENGTH    CVDIG1
## AGE         1.000000000 -0.001190298 0.1946258
## ARMLENGTH -0.001190298  1.000000000 0.9066844
## CVDIG1      0.194625831  0.906684434 1.0000000
```

```
##                AGE    ARMLENGTH    CVDIG1
## AGE         1.00000000 -0.02109405 0.2027411
## ARMLENGTH -0.02109405  1.00000000 0.8898671
## CVDIG1      0.20274112  0.88986710 1.0000000
```

```
##                AGE    ARMLENGTH    CVDIG1
## AGE         1.00000000 -0.01382224 0.1380549
## ARMLENGTH -0.01382224  1.00000000 0.7356773
## CVDIG1      0.13805493  0.73567731 1.0000000
```

c)

	r.0 <dbl>	r.1 <dbl>	r.2 <dbl>	r.3 <dbl>
r squared	0	0.7918635	0.8409529	0.8409658

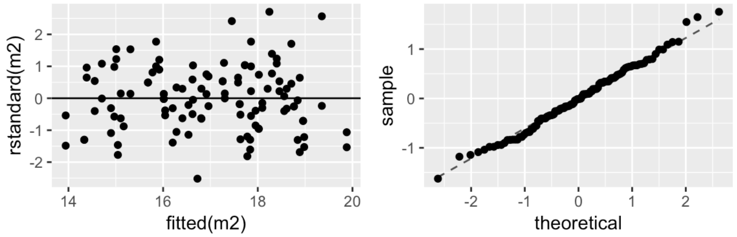
adj r squared	0	0.7899884	0.8380611	0.8365887

2 rows

d)

```
##
## Call:
## lm(formula = CVDIG1 ~ AGE + ARMLENGTH, data = p2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.62499 -0.40716 -0.01396  0.41659  1.75490
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.946019   0.788652  -2.468   0.0151 *
## AGE          0.034890   0.005988   5.827 5.71e-08 ***
## ARMLENGTH    0.196812   0.008368  23.520  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6535 on 110 degrees of freedom
## Multiple R-squared:  0.841,  Adjusted R-squared:  0.8381
## F-statistic: 290.8 on 2 and 110 DF,  p-value: < 2.2e-16
```

e)



f)

	Df <int>	F value <dbl>	Pr(>F) <dbl>
group	58	16.67392	2.996526e-20
	54	*NA*	*NA*

2 rows

	Df <int>	F value <dbl>	Pr(>F) <dbl>
group	58	16.67392	2.996526e-20
	54	*NA*	*NA*

2 rows

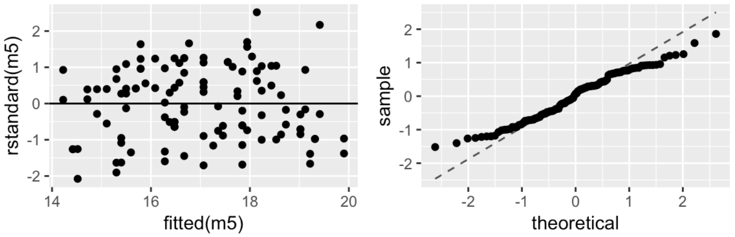
g)

```
##        fit      lwr      upr
## 1 15.20974 14.81428 15.60521
```

```
##        fit      lwr      upr
## 1 15.20974 13.85572 16.56377
```

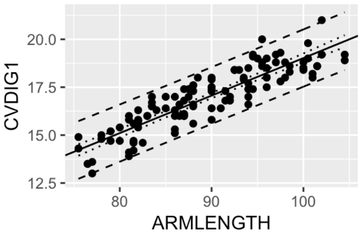
h)

```
##
## Call:
## lm(formula = CVDIG1 ~ ARMLENGTH, data = p2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.51844 -0.62145  0.07524  0.65957  1.85957
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.556875   0.856095   -0.65    0.517
## ARMLENGTH    0.195783   0.009527   20.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7441 on 111 degrees of freedom
## Multiple R-squared:  0.7919, Adjusted R-squared:   0.79
## F-statistic: 422.3 on 1 and 111 DF,  p-value: < 2.2e-16
```

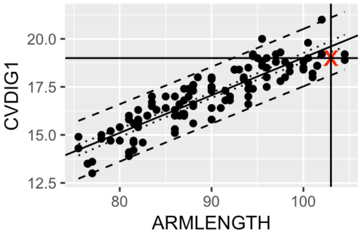


i)

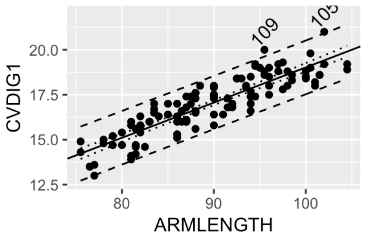
```
## Warning in predict.lm(m5, interval = "predict"): predictions on current data refer
to _future_ responses
```



j)



k)



l)

```
##       fit      lwr      upr
## 1 9.23229 8.472721 9.991858
```

```
##       fit      lwr    upr
## 1 9.23229 7.573577 10.891
```

~~~
---
title: "STA 610/615 Exam 2 Output"
output:
  html_document:
    df_print: paged
  html_notebook: default
---

```{r setup, message=F,warning=F,echo=F}
knitr::opts_chunk$set(
  fig.height=2, fig.width=3, echo=F)
library(mosaic)
library(car)
```
# Problem 1
## a)
```{r rows.print=3}
(read.csv('stroke.csv') %>%
  mutate(SIZE = factor(SIZE,c(2,3,4),
                       c("1-3cm","3-5cm","5cm or greater"))
  ) -> p1)
```

## b)
```{r}
m1 <- lm(SCORES~SIDE*SIZE,p1)
summary(m1)
anova(m1)
```

## c)
```{r}
leveneTest(SCORES~SIDE*SIZE,p1,center="mean")
leveneTest(SCORES~SIDE*SIZE,p1,center="median")
```

## d)
```{r}
gf_point(rstandard(m1)~fitted(m1)) %>% gf_hline(yint=0)
gf_qq(~residuals(m1)) %>% gf_qqline()
filter(p1,SCORES>100)
```

## e)
```{r}
p1a <- mutate(p1,SCORES=ifelse(SCORES>100,19,SCORES))
m5 <- lm(SCORES~SIDE*SIZE,p1a)
anova(m5)
m3 <- lm(SCORES~SIDE+SIZE,p1a)
anova(m3)
```
~~~

~~~
```{r}
leveneTest(SCORES~SIZE,p1a)
gf_point(rstandard(m4)~fitted(m4))  %>% gf_hline(yint=0)
gf_qq(~residuals(m4)) %>% gf_qqline()
```

## h)
```{r}
kruskal.test(SCORES~SIDE,p1a)
kruskal.test(SCORES~SIZE,p1a)
```
## i)
```{r}
with(p1a,pairwise.t.test(SCORES,SIZE,p.adjust.method = 'bonferroni'))
with(p1a,pairwise.wilcox.test(SCORES,SIZE,p.adjust.method = 'bonferroni'))
```
# Problem 2
## a)
```{r fig.height=4, fig.width=6}
p2 <- read.csv('CvDig.csv')
scatterplotMatrix(p2)
```

## b)
```{r}
cor(p2,method='spearman')
cor(p2,method='pearson')
cor(p2,method='kendall')
```

## c)
```{r}
m3 <- lm(CVDIG1~AGE+I(AGE^2)+ARMLENGTH,data=p2)
m2 <- lm(CVDIG1~AGE+ARMLENGTH,data=p2)
m1 <- lm(CVDIG1~ARMLENGTH,data=p2)
m0 <- lm(CVDIG1~1,data=p2)
data.frame(
  r.0 = c(summary(m0)$r.squared,summary(m0)$adj.r.squared),
  r.1 = c(summary(m1)$r.squared,summary(m1)$adj.r.squared),
  r.2 = c(summary(m2)$r.squared,summary(m2)$adj.r.squared),
  r.3 = c(summary(m3)$r.squared,summary(m3)$adj.r.squared),
  row.names = c('r squared','adj r squared')
)
```

## d)
```{r}
summary(m2)
```

## e)
```{r}
~~~

~~~
leveneTest(CVDIG1~factor(AGE)*factor(ARMLENGTH),p2,center="mean")
```

## g)
```{r}
predict(m2,new=data.frame(AGE=63,ARMLENGTH=76),int="conf")
predict(m2,new=data.frame(AGE=63,ARMLENGTH=76),int="pred")
```

## h)
```{r}
m5 <- lm(CVDIG1~ARMLENGTH,data=p2)
summary(m5)
gf_point(rstandard(m5)~fitted(m5)) %>% gf_hline(yint=0)
gf_qq(~residuals(m5)) %>% gf_qqline()
```

## i)
```{r}
m5.a <- predict(m5,interval = 'predict')
m5.b <- predict(m5,interval = 'confidence')
p2 %>%
  gf_point(CVDIG1~ARMLENGTH) %>%
  gf_coefline(model=m5) %>%
  gf_line(m5.a[,2]~ARMLENGTH,lty=2) %>%
  gf_line(m5.a[,3]~ARMLENGTH,lty=2) %>%
  gf_line(m5.b[,2]~ARMLENGTH,lty=3) %>%
  gf_line(m5.b[,3]~ARMLENGTH,lty=3) -> m5.plot
m5.plot
```

## j)
```{r}
m5.plot %>%
  gf_point(19~103,pch='X',col='red',size=4) %>%
  gf_hline(yintercept=19) %>%
  gf_vline(xintercept=103)
```

## k)
```{r}
ip <- which(rstandard(m5)>2)
m5.plot %>%
  gf_text((CVDIG1+1)~ARMLENGTH,
          data=p2[which(rstandard(m5)>2),],
          label=~ip,angle=45)
```

## l)
```{r}
predict(m5,new=data.frame(ARMLENGTH=50),int="conf")
predict(m5,new=data.frame(ARMLENGTH=50),int="pred")
~~~