

W4 practice

2023-02-01

0. data step

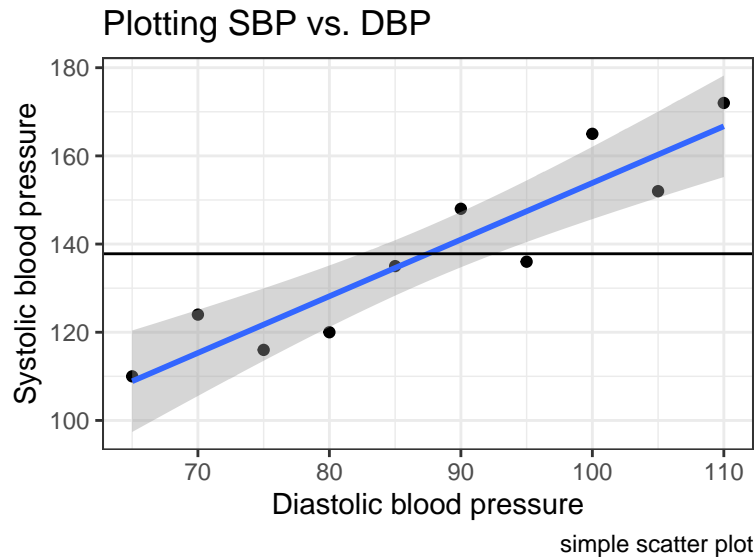
```
dat = data.frame(Patient = c(1:10),  
                 SBP = c(110, 124, 116, 120, 135, 148, 136, 165, 152, 172),  
                 DBP = c( 65, 70, 75, 80, 85, 90, 95, 100, 105, 110))
```

1. Scatterplot

```
tmp.plot =  
dat %>% ggplot(aes(x = DBP, y = SBP)) +  
  geom_point() +  
  xlab("Diastolic blood pressure") + # x-axis label  
  ylab("Systolic blood pressure") + # y-axis label  
  labs(title = "Plotting SBP vs. DBP", caption = "simple scatter plot") +  
  theme_bw()
```

2. adding regression line

```
tmp.plot + # adding to the plot above  
  geom_smooth(method='lm', formula= y ~ x) + # regression line  
  geom_hline(yintercept = mean(dat$SBP)) # mean line
```



3. linear regression

```
fit = lm(SBP ~ DBP, data = dat) # fitting a regression model
summary(fit)
```

```
##
## Call:
## lm(formula = SBP ~ DBP, data = dat)
##
## Residuals:
```

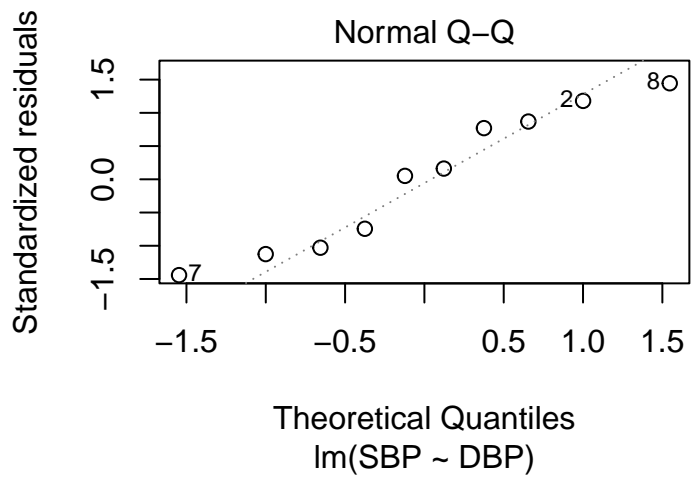
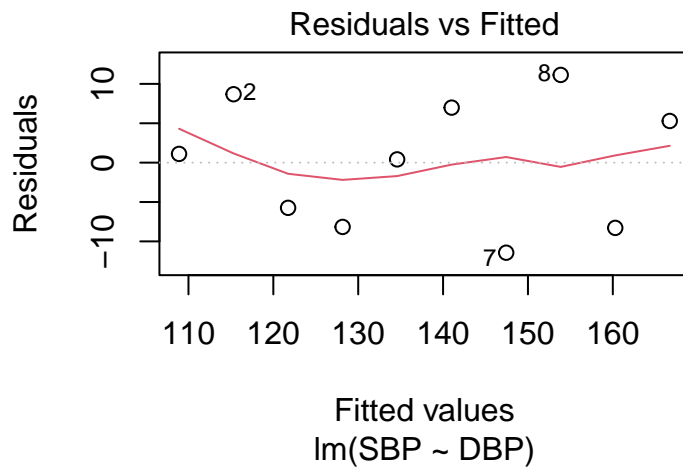
	Min	1Q	Median	3Q	Max
	-11.4364	-7.5576	0.7606	6.5636	11.1394

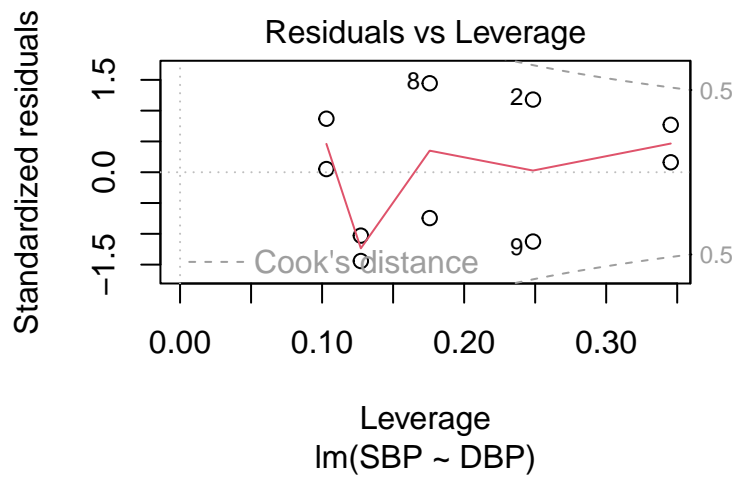
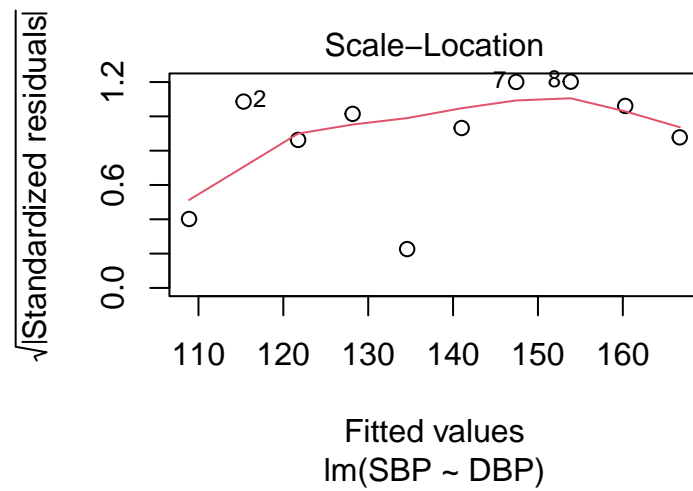
```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	25.376	16.578	1.531	0.164379
DBP	1.285	0.187	6.872	0.000128 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.491 on 8 degrees of freedom
## Multiple R-squared:  0.8551, Adjusted R-squared:  0.837
## F-statistic: 47.23 on 1 and 8 DF, p-value: 0.0001281
```

```
plot(fit) # diagnostic plots
```





```
library(jtools) # summary tool
summ(fit)
```

```
## MODEL INFO:
## Observations: 10
## Dependent Variable: SBP
## Type: OLS linear regression
##
## MODEL FIT:
## F(1,8) = 47.23, p = 0.00
## R2 = 0.86
## Adj. R2 = 0.84
##
## Standard errors: OLS
## -----
```

```
##               Est.    S.E.   t val.    p
## -----
## (Intercept)    25.38   16.58     1.53   0.16
## DBP            1.28    0.19     6.87   0.00
## -----
```

```
summ(fit, scale = TRUE) # gives standardized coefficients
```

```
## MODEL INFO:
## Observations: 10
## Dependent Variable: SBP
## Type: OLS linear regression
##
## MODEL FIT:
## F(1,8) = 47.23, p = 0.00
## R2 = 0.86
## Adj. R2 = 0.84
##
## Standard errors: OLS
## -----
##               Est.    S.E.   t val.    p
## -----
## (Intercept)    137.80   2.69    51.32   0.00
## DBP            19.45   2.83     6.87   0.00
## -----
##
## Continuous predictors are mean-centered and scaled by 1 s.d.
```

```
summ(fit, confint = TRUE, digits = 3) # with confidence intervals
```

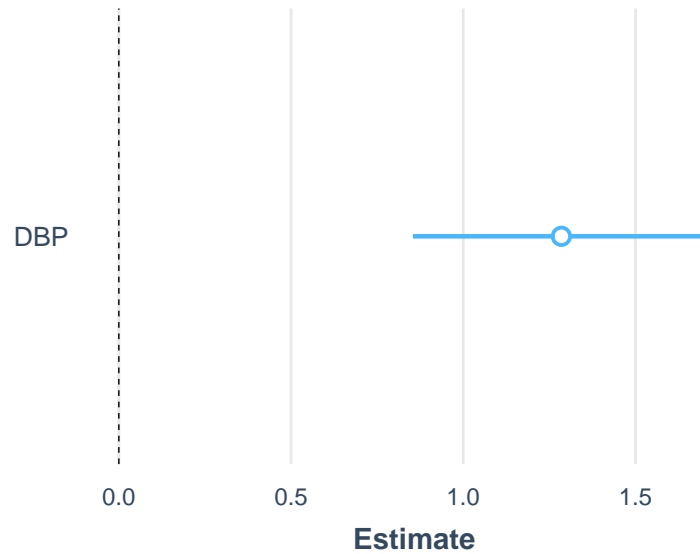
```
## MODEL INFO:
## Observations: 10
## Dependent Variable: SBP
## Type: OLS linear regression
##
## MODEL FIT:
## F(1,8) = 47.228, p = 0.000
## R2 = 0.855
## Adj. R2 = 0.837
##
## Standard errors: OLS
## -----
##               Est.    2.5%   97.5%   t val.    p
## -----
## (Intercept)    25.376  -12.853  63.605    1.531   0.164
## DBP            1.285    0.854   1.716    6.872   0.000
## -----
```

```
plot_summs(fit) # coefficient plot
```

```
## Registered S3 methods overwritten by 'broom':
##   method          from
```

```
## tidy.glht jtools
## tidy.summary.glht jtools

## Loading required namespace: broom.mixed
```



```
export_summs(fit) # table
```

	Model 1
(Intercept)	25.38 (16.58)
DBP	1.28 *** (0.19)
N	10
R2	0.86

*** p < 0.001; ** p < 0.01; * p < 0.05.

4. ANOVA paw lift

```
rat = data.frame(rat_id = c(1:72),
                 drug = factor(c(rep(c("A", "B", "C"), each = 8, times = 3))),
                 timelift = c(5.2, 5.1, 2.4, 4.3, 3.1, 2.4, 4.2, 5.0, 2.4, 2.4, 4.6, 1.2, 5.1, 5.2, 2.9,
                              5.2, 5.1, 2.4, 4.3, 3.1, 2.4, 4.2, 5.0, 2.4, 2.4, 4.6, 1.2, 5.1, 5.2, 2.9,
                              5.2, 5.1, 2.4, 4.3, 3.1, 2.4, 4.2, 5.0, 2.4, 2.4, 4.6, 1.2, 5.1, 5.2, 2.9)),
                 stringsAsFactors = FALSE)

describe(rat)
```

```
##          vars  n mean    sd median trimmed   mad min  max range skew kurtosis
## rat_id      1 72 36.50 20.93   36.5   36.50 26.69 1.0 72.0    71    0   -1.25
## drug*       2 72  2.00  0.82    2.0    2.00  1.48 1.0  3.0     2    0   -1.54
## timelift    3 72  3.52  0.93    3.5    3.51  0.96 1.2  5.2     4    0   -0.68
##          se
## rat_id     2.47
## drug*      0.10
## timelift   0.11
```

```
describeBy(rat, "drug") # descriptive stats by drug
```

```
##
## Descriptive statistics by group
## drug: A
##          vars  n mean    sd median trimmed   mad min  max range skew kurtosis
## rat_id      1 24 28.50 20.15   28.5   28.50 32.62 1.0 56.0   55.0  0.00   -1.59
## drug*       2 24  1.00  0.00    1.0    1.00  0.00 1.0  1.0     0.0   NaN    NaN
## timelift    3 24  3.64  1.04    3.8    3.65  1.19 1.8  5.2     3.4 -0.17   -1.29
##          se
## rat_id     4.11
## drug*      0.00
## timelift   0.21
## -----
## drug: B
##          vars  n mean    sd median trimmed   mad min  max range skew kurtosis
## rat_id      1 24 36.50 20.15   36.5   36.50 32.62 9.0 64.0    55  0.00   -1.59
## drug*       2 24  2.00  0.00    2.0    2.00  0.00 2.0  2.0     0   NaN    NaN
## timelift    3 24  3.52  0.89    3.6    3.54  0.59 1.2  5.2     4 -0.33    0.32
##          se
## rat_id     4.11
## drug*      0.00
## timelift   0.18
## -----
## drug: C
##          vars  n mean    sd median trimmed   mad min  max range skew kurtosis
## rat_id      1 24 44.50 20.15   44.50   44.50 32.62 17 72.0   55.0  0.00   -1.59
## drug*       2 24  3.00  0.00    3.00    3.00  0.00  3  3.0     0.0   NaN    NaN
## timelift    3 24  3.40  0.86    3.15    3.36  0.67  2  5.1     3.1  0.48   -0.95
##          se
## rat_id     4.11
## drug*      0.00
## timelift   0.18
```

```
rat %>% filter(drug == "A") %>% describe
```

```
##          vars  n mean    sd median trimmed   mad min  max range skew kurtosis
## rat_id      1 24 28.50 20.15   28.5   28.50 32.62 1.0 56.0   55.0  0.00   -1.59
## drug*       2 24  1.00  0.00    1.0    1.00  0.00 1.0  1.0     0.0   NaN    NaN
## timelift    3 24  3.64  1.04    3.8    3.65  1.19 1.8  5.2     3.4 -0.17   -1.29
##          se
## rat_id     4.11
## drug*      0.00
## timelift   0.21
```

```
fit2 = lm(timelift ~ drug, data = rat) # regression
summary(fit2)
```

```
##
## Call:
## lm(formula = timelift ~ drug, data = rat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3250 -0.5958 -0.0250  0.6656  1.7042
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6375     0.1908  19.064 <2e-16 ***
## drugB         -0.1125     0.2698  -0.417  0.678
## drugC         -0.2417     0.2698  -0.896  0.374
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9348 on 69 degrees of freedom
## Multiple R-squared:  0.01151,    Adjusted R-squared:  -0.01714
## F-statistic: 0.4017 on 2 and 69 DF,  p-value: 0.6708
```

```
rat$drug = releval(rat$drug, ref = c("B"))
```

```
library(multcomp)
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

```
## Loading required package: MASS
```

```
##
```

```
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      select
```

```
##
```

```
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
```

```
##
```

```
##      geyser
```

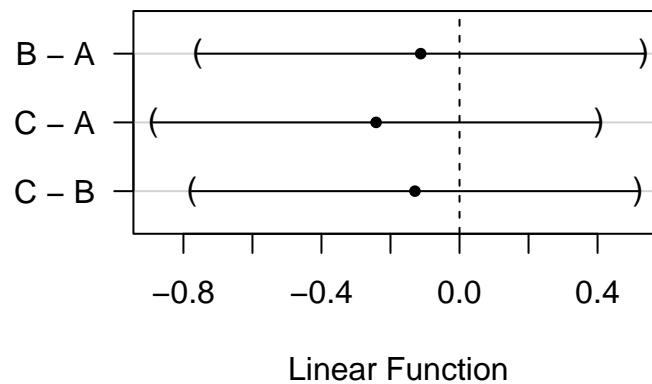


```
tmp.comp = glht(fit2, linfct = mcp(drug = "Tukey")) # multiple comparison with Tukey method
summary(tmp.comp)
```

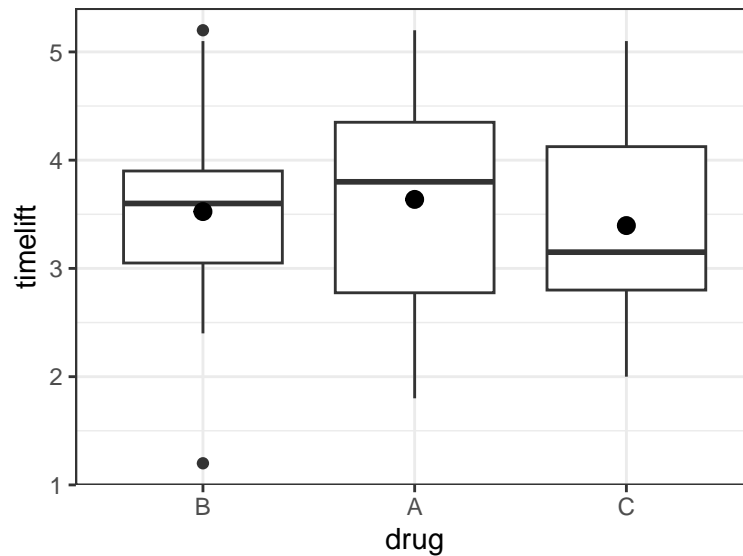
```
##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = timelift ~ drug, data = rat)
##
## Linear Hypotheses:
##           Estimate Std. Error t value Pr(>|t|)
## B - A == 0  -0.1125     0.2698  -0.417   0.909
## C - A == 0  -0.2417     0.2698  -0.896   0.645
## C - B == 0  -0.1292     0.2698  -0.479   0.882
## (Adjusted p values reported -- single-step method)
```

```
plot(tmp.comp)
```

95% family-wise confidence level



```
rat %>%
  ggplot(aes(x=drug, y=timelift)) + # create a boxplot
  geom_boxplot() +
  stat_summary(fun.y="mean") + # draw mean
  theme_bw()
```



5. Create indicator variables

```
ratsreg =
rat %>% mutate(drugA = ifelse(drug == "A", 1, 0),
               drugB = ifelse(drug == "B", 1, 0),
               drugC = ifelse(drug == "C", 1, 0))

table(ratsreg$drug, ratsreg$drugA)
```

```
##
##      0  1
##   B 24  0
##   A  0 24
##   C 24  0
```

```
table(ratsreg$drug, ratsreg$drugB)
```

```
##
##      0  1
##   B  0 24
##   A 24  0
##   C 24  0
```

```
fit3 = lm(timelift ~ drugA + drugB, data = ratsreg)
summary(fit3)
```

```
##
## Call:
## lm(formula = timelift ~ drugA + drugB, data = ratsreg)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3250 -0.5958 -0.0250  0.6656  1.7042
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.3958     0.1908  17.797  <2e-16 ***
## drugA         0.2417     0.2698   0.896   0.374
## drugB         0.1292     0.2698   0.479   0.634
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9348 on 69 degrees of freedom
## Multiple R-squared:  0.01151,    Adjusted R-squared:  -0.01714
## F-statistic: 0.4017 on 2 and 69 DF,  p-value: 0.6708
```

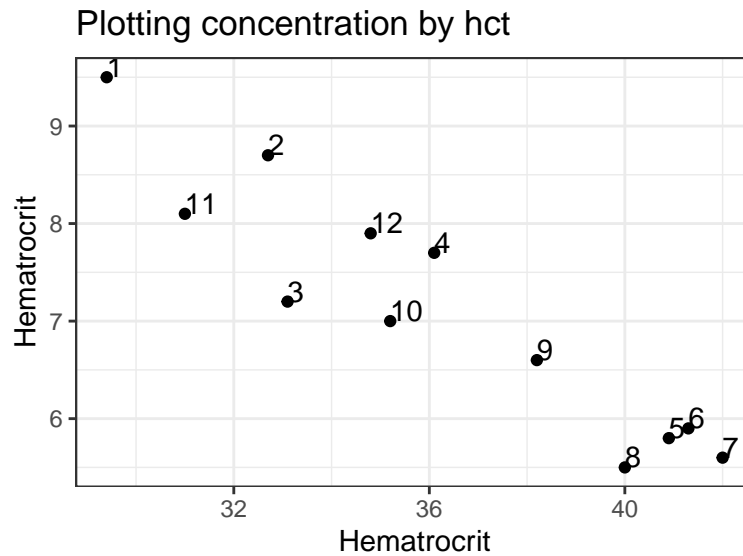
6. Example 3. Gentamicin

```
lect4a = data.frame(id = c(1:12),
                    hgb = c(9.7, 10.9, 11.3, 12.0, 13.5, 13.7, 14.2, 13.2, 12.5, 11.7, 10.5, 11.2),
                    hct = c(29.4, 32.7, 33.1, 36.1, 40.9, 41.3, 42.0, 40.0, 38.2, 35.2, 31.0, 34.8),
                    concn = c(9.5, 8.7, 7.2, 7.7, 5.8, 5.9, 5.6, 5.5, 6.6, 7.0, 8.1, 7.9))
```

```
lect4a %>% describe
```

```
##      vars  n mean  sd median trimmed  mad  min  max range  skew kurtosis
## id       1 12  6.50 3.61   6.50    6.50 4.45   1.0 12.0  11.0  0.00   -1.50
## hgb      2 12 12.03 1.40  11.85   12.05 1.70   9.7 14.2   4.5  0.03   -1.41
## hct      3 12 36.23 4.25  35.65   36.33 5.41  29.4 42.0  12.6 -0.06   -1.55
## concn    4 12  7.12 1.30   7.10    7.05 1.63   5.5  9.5   4.0  0.26   -1.32
##
##      se
## id    1.04
## hgb   0.40
## hct   1.23
## concn 0.37
```

```
lect4a %>%
  ggplot(aes(x=hct, y=concn, label=id)) +
  geom_point() +
  geom_text(hjust=0, vjust=0) +
  xlab("Hematocrit") +
  ylab("Hematocrit") +
  labs(title = "Plotting concentration by hct") +
  theme_bw()
```

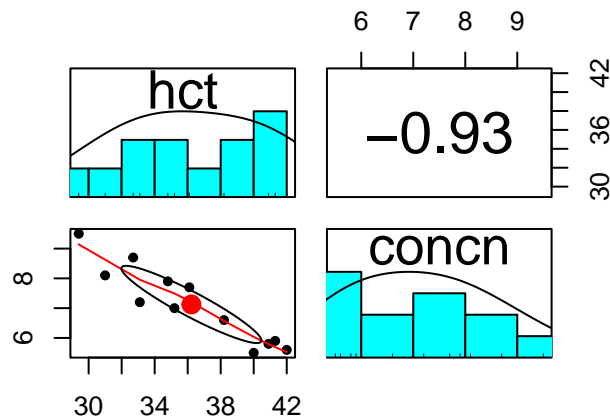


7. correlation

```
corr.test(lect4a[,c('hct', 'concn')]) # coefficient and p-values
```

```
## Call:corr.test(x = lect4a[, c("hct", "concn")])
## Correlation matrix
##      hct concn
## hct   1.00 -0.93
## concn -0.93  1.00
## Sample Size
## [1] 12
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##      hct concn
## hct    0     0
## concn  0     0
##
## To see confidence intervals of the correlations, print with the short=FALSE option
```

```
pairs.panels(lect4a[,c('hct', 'concn')]) # more visuals
```



```
lowerCor(lect4a[,c('hct', 'concn')]) # lower correlation plot
```

```
##      hct  concn
## hct    1.00
## concn -0.93  1.00
```

```
fit4 = lm(concn ~ hct, data = lect4a)
summary(fit4)
```

```
##
## Call:
## lm(formula = concn ~ hct, data = lect4a)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.81589 -0.44155  0.07971  0.38389  0.57007
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.45223    1.27014   13.740 8.10e-08 ***
## hct          -0.28509    0.03484   -8.182 9.67e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4906 on 10 degrees of freedom
## Multiple R-squared:  0.87, Adjusted R-squared:  0.857
## F-statistic: 66.94 on 1 and 10 DF, p-value: 9.666e-06
```

```
anova(fit4) #anova
```

```
## Analysis of Variance Table
##
## Response: concn
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## hct         1 16.1152 16.1152  66.942 9.666e-06 ***
## Residuals   10  2.4073  0.2407
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lect4a %>% ggplot(aes(x=hct, y=concn)) +
  geom_point() +
  geom_smooth(method='lm', formula= y~x) +
  theme_bw()
```

