BST 210 Lab: Week 1

Introduction and Simple Linear Regression

This week, we will get started with the very basics of R, SAS and Stata. We'll talk about how to install or get access to those softwares in your computer, and then repeat a simple data analysis with three different languages. We do not assume any prior knowledge of R, SAS or Stata, but will expect a general understanding of some programming language/syntax (or at least the willingness to learn). Throughout the remaining lectures, examples will be presented using any one of the three specified softwares. It is our hope that after understanding the material, you could replicate a certain example using any of these softwares. At the end of this lab session, we'll go through simple linear regression basics and analyze a new dataset.

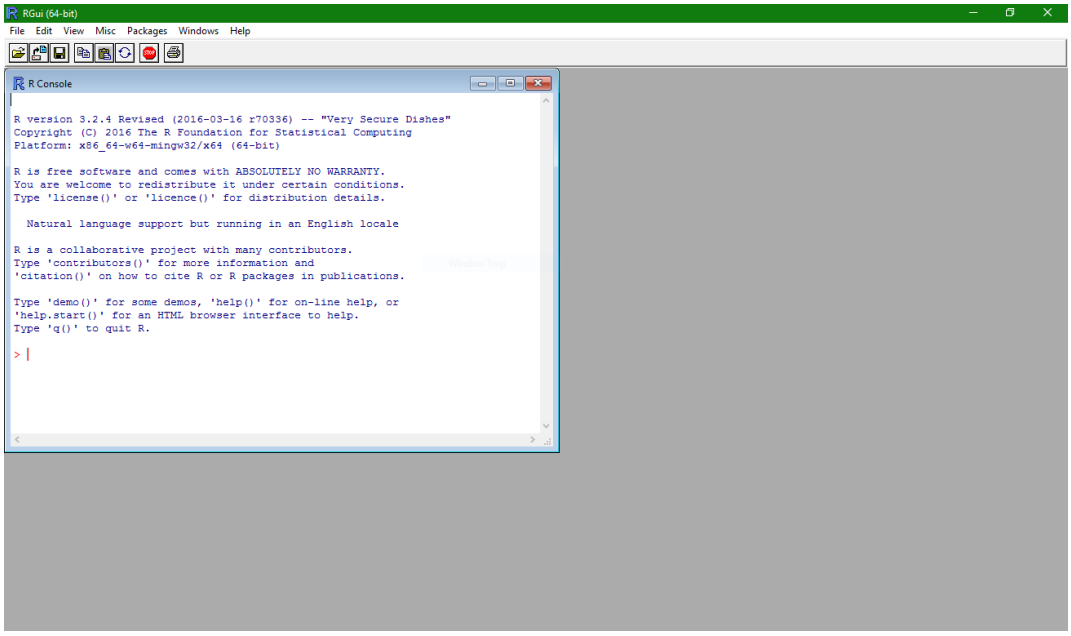
Getting Started with R

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

Downloading R

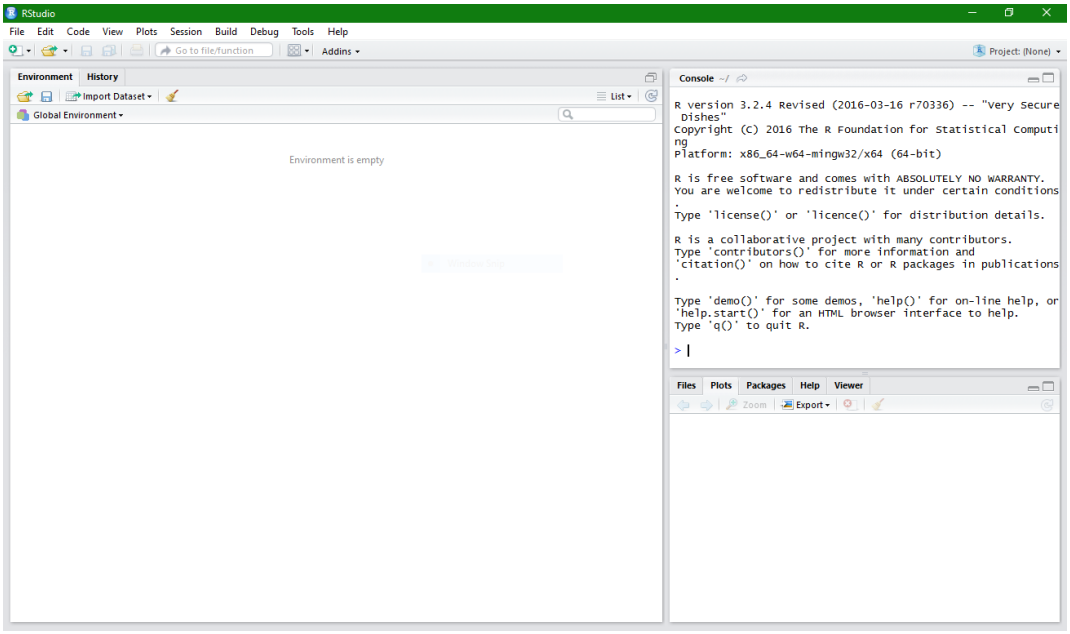
1. Visit `https://www.r-project.org/` in any browser.
2. Click **download R**.
3. Choose a mirror, any will do, but closer mirrors geographically should theoretically download faster.
4. Choose the appropriate R for your operating system (Windows, Mac, Linux all supported).
5. Click **base**.
6. Click **Download R...**.
7. Run installer with default options.
8. Congratulations! You can now open and use R!



Downloading RStudio

Although you can do just about everything in R, most find it easier to work with an Integrated Development Environment, which will allow you to write and test R scripts more easily. RStudio is an IDE for R that can be downloaded freely for personal use.

1. Visit `https://www.rstudio.com/`.
2. Click **Download** under **Rstudio** logo.
3. Choose the **FREE** version and click **DOWNLOAD**.
4. Choose the appropriate installer based on your own operating system.
5. Run the installer using default settings.
6. Congratulations! You can now open and use RStudio!



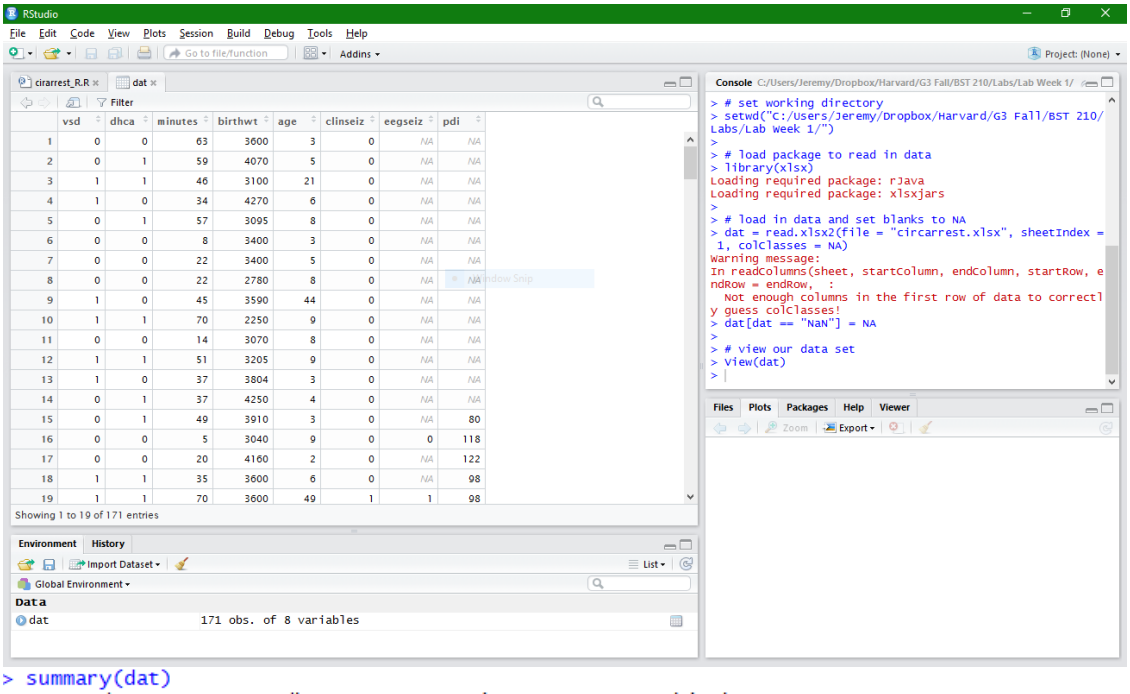
Computing in R

Now we're going to run through the circarrest examples presented in class, but this time, using R! Download the "Data and Programs" folder for Lab 1 from Canvas.

Loading in the data

Open 'circarreststart.R' file or follow the scripts below. The command `setwd` stands for "Set working directory." You'll need to adjust the directory to where you stored the data. On a Mac it might look something like `setwd("~/Downloads/Data and Programs/")`.

```
# set working directory
setwd("C:/Users/glius/Dropbox/BST 210/Labs (Fall 2017)/Week 1")
dat = read.csv("circarrest.csv",header = T)

# view our data set
View(dat)

# check summary statistics
summary(dat)
```




```
> summary(dat)
      vsd              dhca            minutes          birthwt         age
 Min.   :0.0000   Min.   :0.0000   Min.   :  3.00   Min.   :2250   Min.   : 1.000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 13.00   1st Qu.:3235   1st Qu.: 4.000
 Median :0.0000   Median :1.0000   Median : 41.00   Median :3500   Median : 6.000
 Mean   :0.2456   Mean   :0.5088   Mean   : 35.98   Mean   :3534   Mean   : 9.901
 3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.: 55.00   3rd Qu.:3842   3rd Qu.: 9.000
 Max.   :1.0000   Max.   :1.0000   Max.   :103.00   Max.   :4600   Max.   :67.000

    clinseiz          eegseiz           pdi
 Min.   :0.00000   Min.   :0.0000   Min.   : 52.0
 1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.: 86.0
 Median :0.00000   Median :0.0000   Median : 98.0
 Mean   :0.06471   Mean   :0.1985   Mean   : 95.1
 3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:105.0
 Max.   :1.00000   Max.   :1.0000   Max.   :134.0
 NA's   :1         NA's   :35       NA's   :29
```

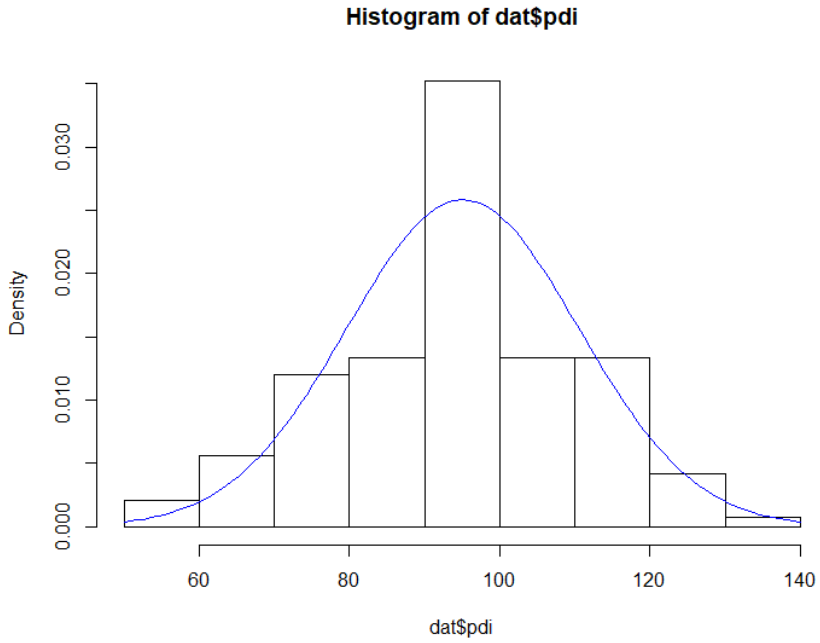
Histograms

```
# histogram of pdi
hist(dat$pdi,prob=T)

# add on a normal overlay - requires a mean/standard deviation
pdi_mean = mean(dat$pdi, na.rm = T)
pdi_std  = sqrt(var(dat$pdi, na.rm = T))
curve(dnorm(x, mean=pdi_mean, sd=pdi_std),
col="blue", add=T)
```

Note:

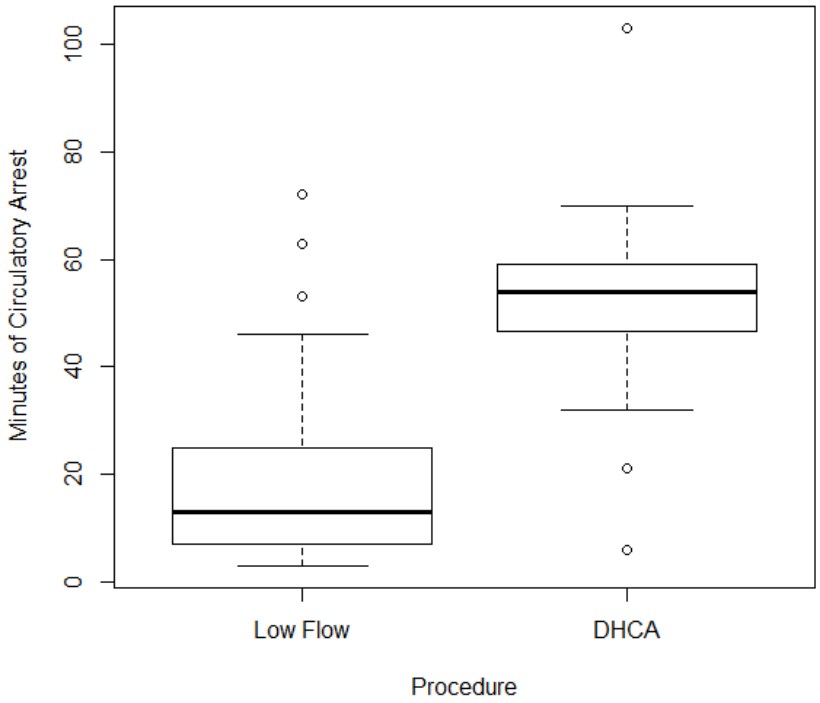
1.'prob=T' option in 'hist' function means the histogram graphic is a representation of probability densities. Otherwise, 'prob=F' means the histogram graphic represents frequency, the counts component of the result.
2. You can choose the number of bins using commands like:

```
hist(dat$pdi,seq(min(dat$pdi,na.rm = T),max(dat$pdi,na.rm = T),length.out=10))
```



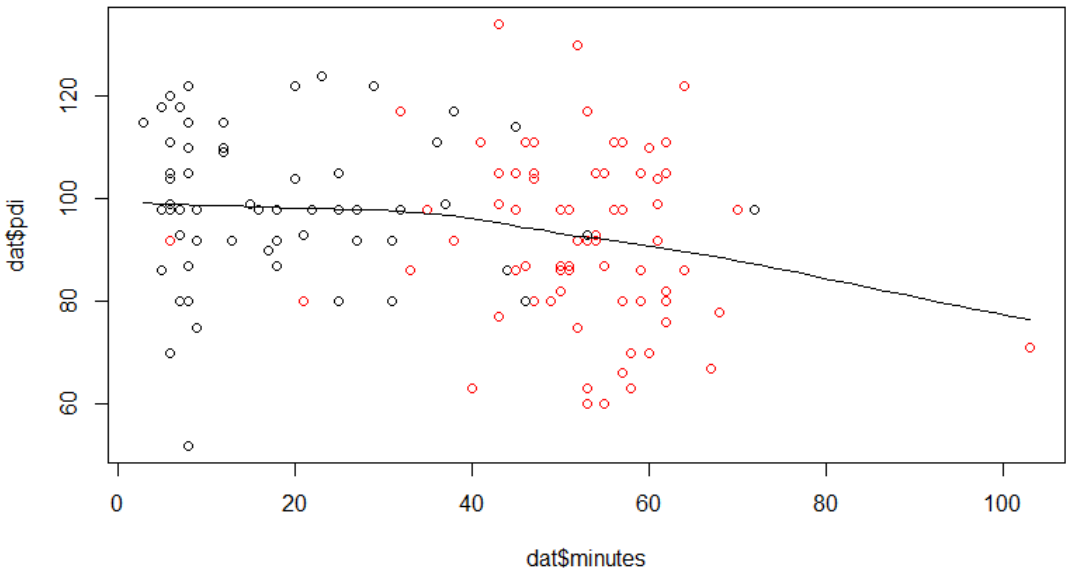
Boxplots

```
# box plot of minutes of circarrest by procedure type
boxplot(dat$minutes ~ factor(dat$dhca, levels = c(0,1),labels = c("Low Flow", "DHCA")),
         na.rm = TRUE,
        xlab = "Procedure",
        ylab = "Minutes of Circulatory Arrest")
```



Scatter Plots and LOESS curves

```
# scatter plot
plot(dat$minutes, dat$pdi, col = dat$dhca + 1)
# loess curve
scatter.smooth(dat$minutes, dat$pdi, col = dat$dhca + 1)
```



Simple Hypothesis Tests

```
> # t-test
> t.test(dat$pdi, mu = 110)

One Sample t-test

data:  dat$pdi
t = -11.493, df = 141, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 110
95 percent confidence interval:
92.53537 97.66181
sample estimates:
mean of x
95.09859
```

```
> t.test(dat$pdi, mu = 110, alternative = "greater")

One Sample t-test

data:  dat$pdi
t = -11.493, df = 141, p-value = 1
alternative hypothesis: true mean is greater than 110
95 percent confidence interval:
```

```
92.95183      Inf
sample estimates:
mean of x
95.09859
```

```
> t.test(dat$pdi, mu = 110, alternative = "less")

One Sample t-test

data:  dat$pdi
t = -11.493, df = 141, p-value < 2.2e-16
alternative hypothesis: true mean is less than 110
95 percent confidence interval:
-Inf 97.24536
sample estimates:
mean of x
95.09859
```

```
> # Pearson's correlation test
> cor.test(dat$pdi, dat$minutes)

Pearson's product-moment correlation

data:  dat$pdi and dat$minutes
t = -2.6654, df = 140, p-value = 0.008595
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.37105202 -0.05709641
sample estimates:
cor
-0.2197568
```

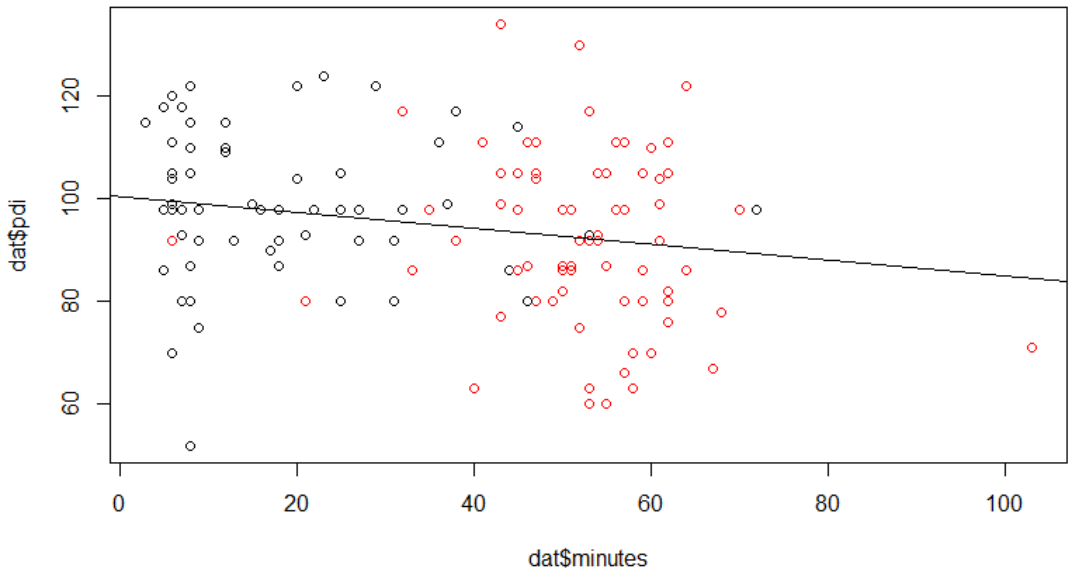
```
> # two-sample t-test, default in R is to assume unequal variance
> t.test(dat$pdi ~ dat$dhca, var.equal = T)

Two Sample t-test

data:  dat$pdi by dat$dhca
t = 2.5732, df = 140, p-value = 0.01112
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
1.516439 11.575480
sample estimates:
mean in group 0 mean in group 1
98.46377        91.91781
```

Linear Regression

```
> # Linear Regression
> mod1 = lm(pdi ~ minutes, data = dat)
> summary(mod1)

Call:
lm(formula = pdi ~ minutes, data = dat)

Residuals:
   Min      1Q  Median      3Q     Max
-47.335  -7.772  -0.459  11.231  40.073

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 100.57084    2.41383  41.664  < 2e-16 ***
minutes      -0.15452    0.05797  -2.665  0.00859 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.13 on 140 degrees of freedom
  (29 observations deleted due to missingness)
Multiple R-squared:  0.04829,        Adjusted R-squared:  0.0415
F-statistic: 7.104 on 1 and 140 DF,  p-value: 0.008595

> confint(mod1)
                2.5 %       97.5 %
(Intercept) 95.7985744 105.34311218
minutes     -0.2691295  -0.03990206
```

Plotting a regression line

```
# scatter plot with fitted values
plot(dat$minutes, dat$pdi, col = dat$dhca + 1)
abline(mod1)
```



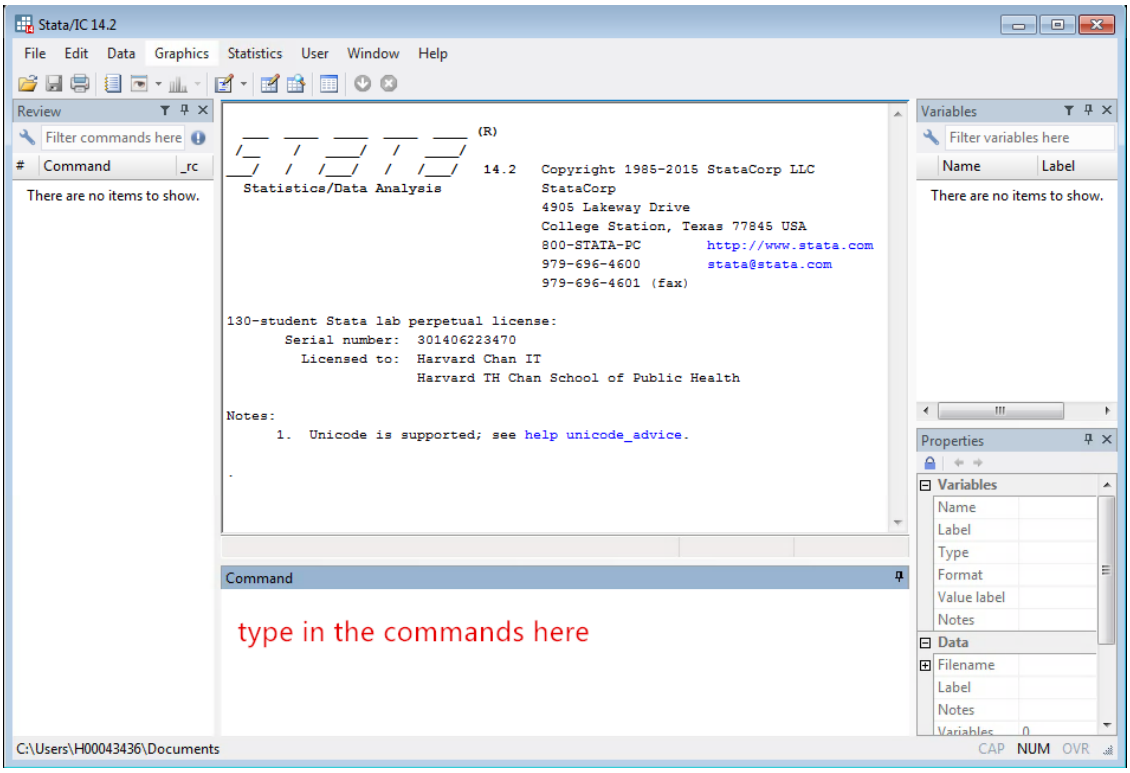
Getting access to SAS and Stata with Virtual Desktop

Downloading VMWare

1. Go to `https://www.hsph.harvard.edu/information-technology/student-computing/virtual-desktop-for-students-vdi/`, follow the instruction and install the Virtual Desktop.
2. Log in the Virtual Desktop with your username and password (for printing)
3. Open SAS/Stata from the Virtual Desktop

Next, we repeat the same analysis we did in R using Stata 14.

Stata 14



Loading in the data

Open 'circarreststart.do' file or follow the scripts below.

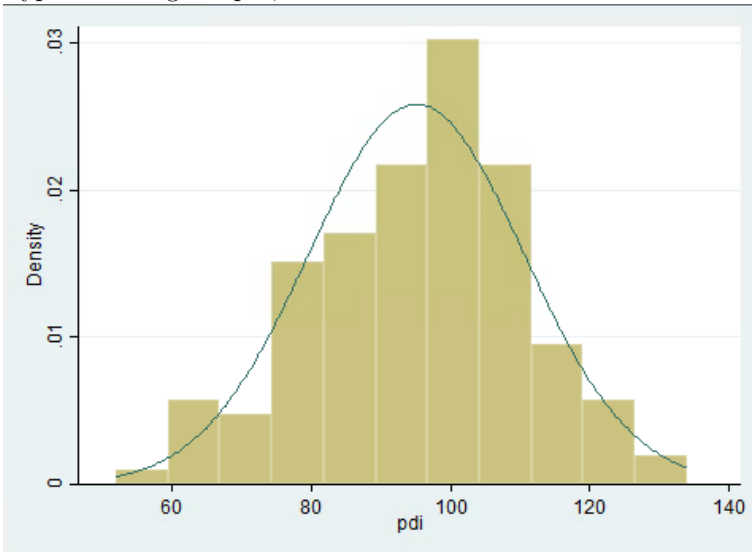
```
# For .dta files, type in
use "P:\Week_1_export\Data and Programs\circarrest.dta"
# For .csv files
1. Open csv file first, press "ctrl+f", click 'replace', fill in 'find what' with NA,
   then click 'replace all'.
2. In Stata, click 'file' tab, choose 'import', choose 'Text data (delimited, csv)',
   find the csv file and click 'OK'.

Then type in 'summarize'.
```

```
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
vsd	171	.245614	.4317148	0	1
dhca	171	.5087719	.5013913	0	1
minutes	171	35.97661	21.7139	3	103
birthwt	171	3533.596	429.6101	2250	4600
age	171	9.900585	11.29011	1	67
clinseiz	170	.0647059	.2467329	0	1
eegseiz	136	.1985294	.4003675	0	1
pdi	142	95.09859	15.45036	52	134

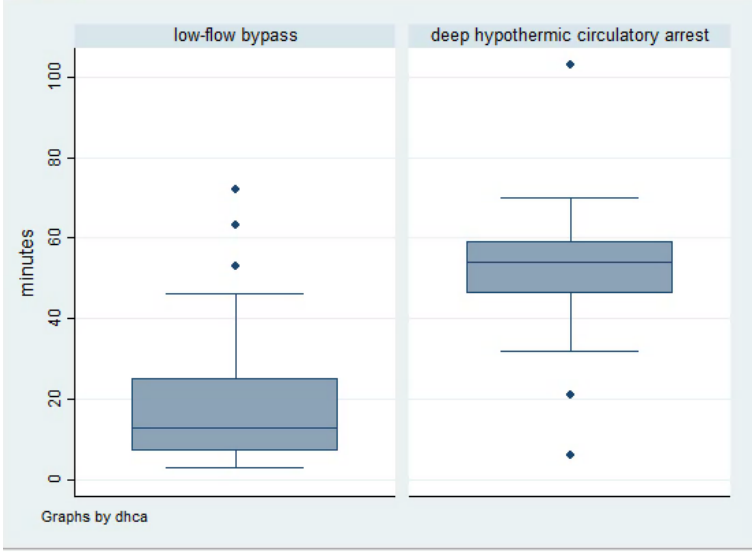
Histograms

Type in 'histogram pdi, normal'



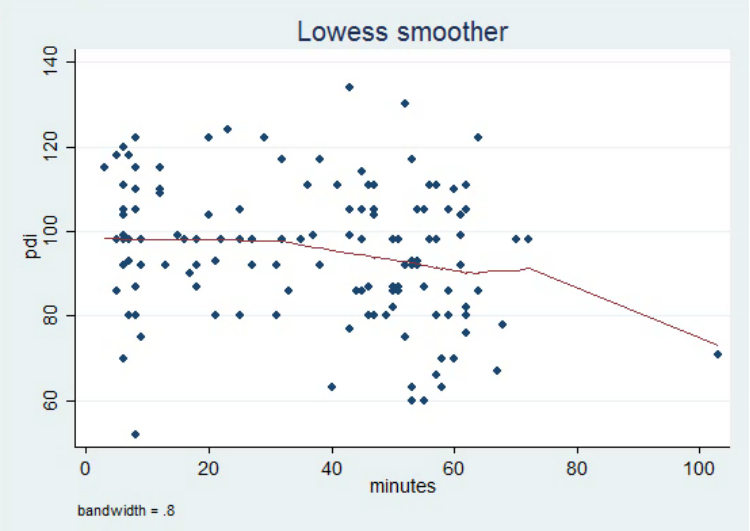
Boxplots

Type in 'graph box minutes, by(dhca)'



Scatter Plots and LOESS curves

```
scatter pdi minutes
lowess pdi minutes
```



Simple Hypothesis Tests

```
# For one-sample T test
ttest pdi==110
```

```
# For two-sample T test
ttest pdi, by(dhca)
# Pearson correlation test
pwcorr pdi minutes, sig
```

```
. ttest pdi==110

One-sample t test

Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
     pdi |     142    95.09859    1.296565    15.45036    92.53537    97.66181

    mean = mean(pdi)                                              t = -11.4930
Ho: mean = 110                                  degrees of freedom =      141

   Ha: mean < 110               Ha: mean != 110                 Ha: mean > 110
 Pr(T < t) = 0.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 1.0000

. ttest pdi, by(dhca)

Two-sample t test with equal variances

   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
low-flow |      69    98.46377    1.636459    13.59345    95.19827    101.7293
deep hyp |      73    91.91781    1.929775      16.488    88.07087    95.76474
---------+--------------------------------------------------------------------
combined |     142    95.09859    1.296565    15.45036    92.53537    97.66181
---------+--------------------------------------------------------------------
    diff |            6.54596    2.543947                1.516439    11.57548

    diff = mean(low-flow) - mean(deep hyp)                        t =   2.5732
Ho: diff = 0                                    degrees of freedom =      140

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.9944         Pr(|T| > |t|) = 0.0111          Pr(T > t) = 0.0056

. pwcorr pdi minutes, sig

             |      pdi  minutes
-------------+------------------
         pdi |   1.0000
             |
             |
     minutes |  -0.2198   1.0000
             |   0.0086
```

Linear Regression

Type in 'regress pdi minutes'

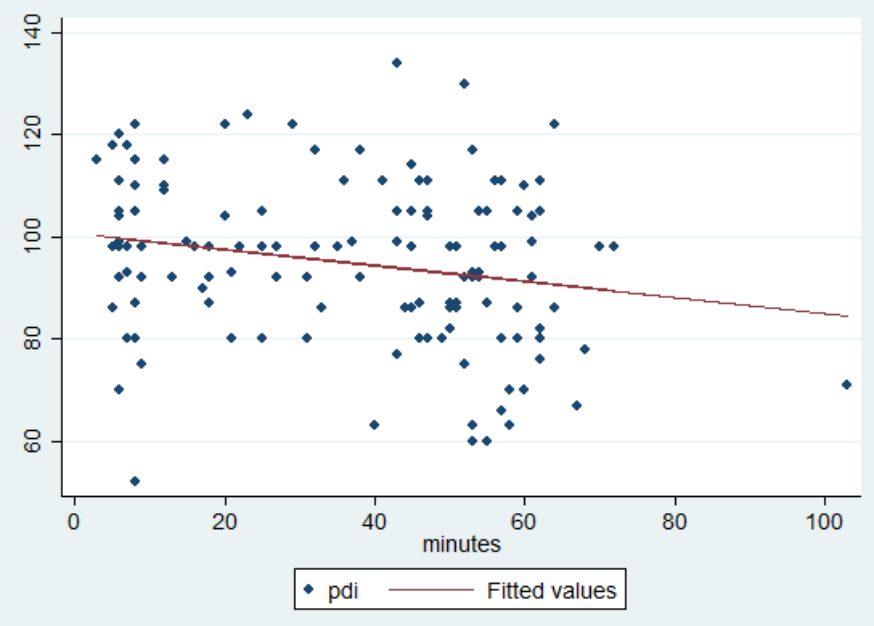
```
. regress pdi minutes
```

Source	SS	df	MS	Number of obs	=	142
				F(1, 140)	=	7.10
Model	1625.47748	1	1625.47748	Prob > F	=	0.0086
Residual	32033.1422	140	228.808159	R-squared	=	0.0483
				Adj R-squared	=	0.0415
Total	33658.6197	141	238.713615	Root MSE	=	15.126

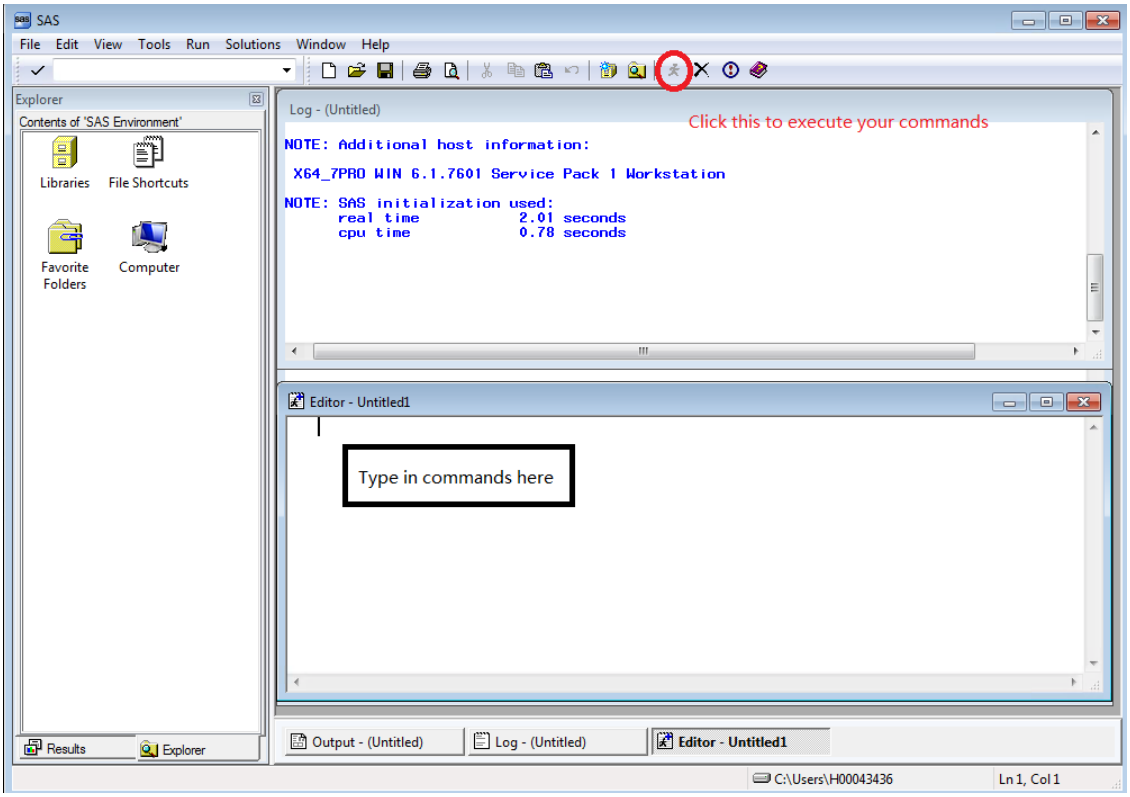
pdi	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
minutes	-.1545158	.057972	-2.67	0.009	-.2691295	-.0399021
_cons	100.5708	2.413829	41.66	0.000	95.79857	105.3431

Plotting a regression line

```
predict pdifitted
twoway scatter pdi minutes || line pdifitted minutes
```



SAS 9.4



Loading in the data

Open 'circarreststart.sas' file or follow the scripts below.

```
# For .xlsx files,
proc import out = dat
datafile = "P:\Week_1_export\Data and Programs\circarrest.xlsx"
dbms = xlsx;
run;
# For .csv files
replace 'dbms=xlsx' with 'dbms=csv'
# Click 'Libraries'-->'Work'-->'Dat' to view the data
# Check summary statistics
proc means data = dat;
run;
```

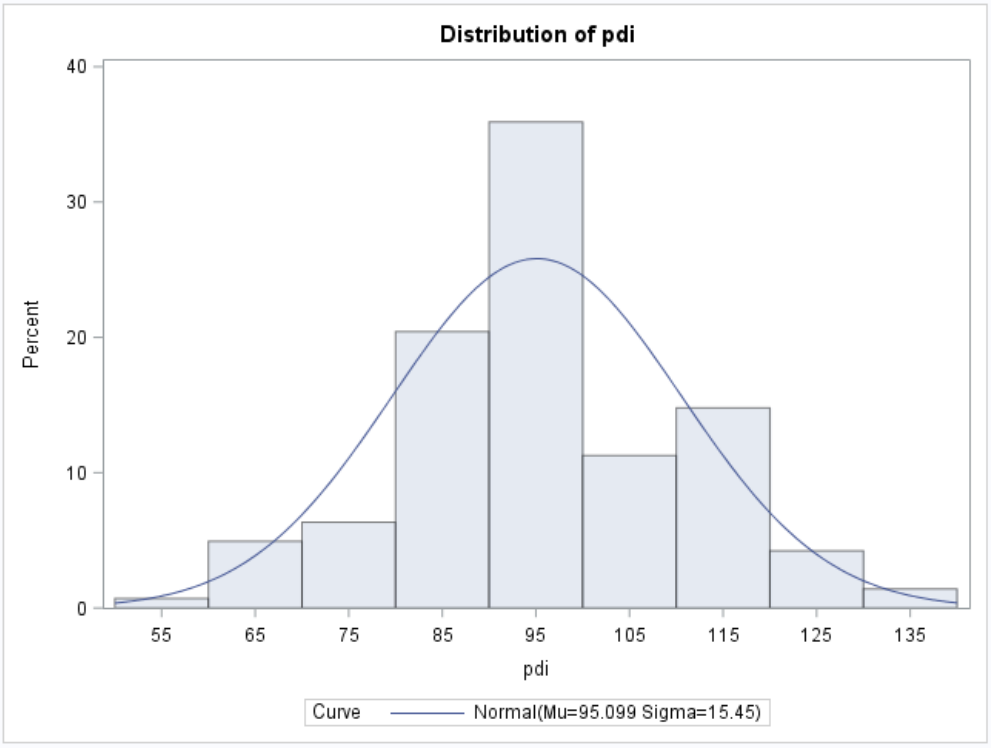
The SAS System

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
vsd	171	0.2456140	0.4317148	0	1.0000000
dhca	171	0.5087719	0.5013913	0	1.0000000
minutes	171	35.9766082	21.7139026	3.0000000	103.0000000
birthwt	171	3533.60	429.6101184	2250.00	4600.00
age	171	9.9005848	11.2901076	1.0000000	67.0000000
clinseiz	170	0.0647059	0.2467329	0	1.0000000
eegseiz	136	0.1985294	0.4003675	0	1.0000000
pdi	142	95.0985915	15.4503597	52.0000000	134.0000000

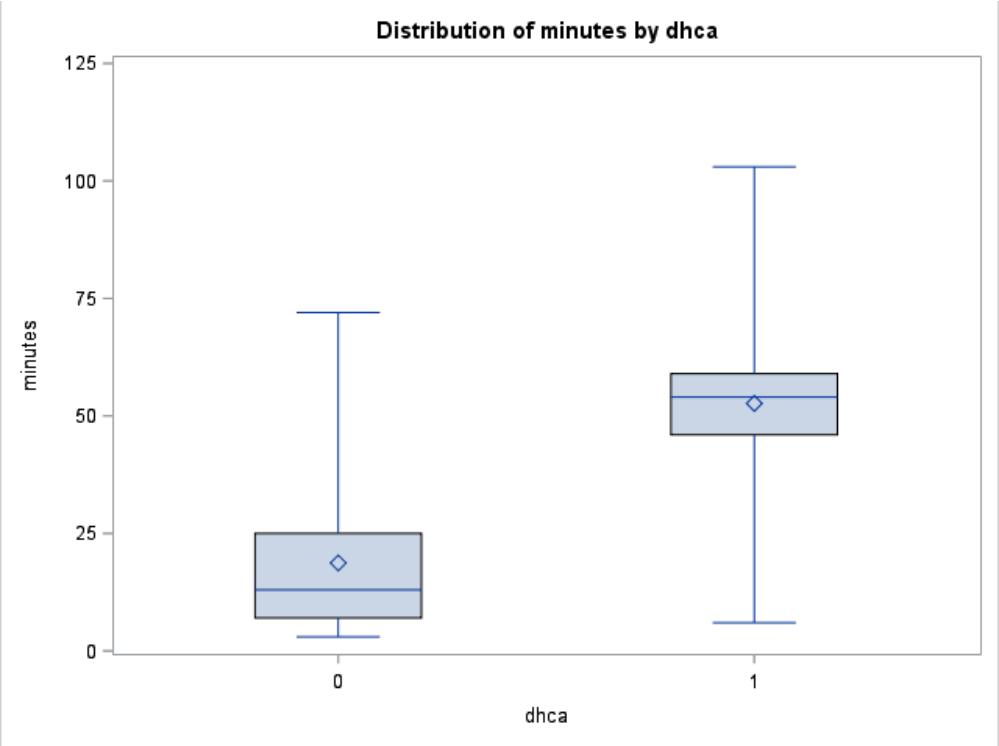
Histograms

```
proc univariate data = dat noprint;
histogram pdi / normal;
run;
```



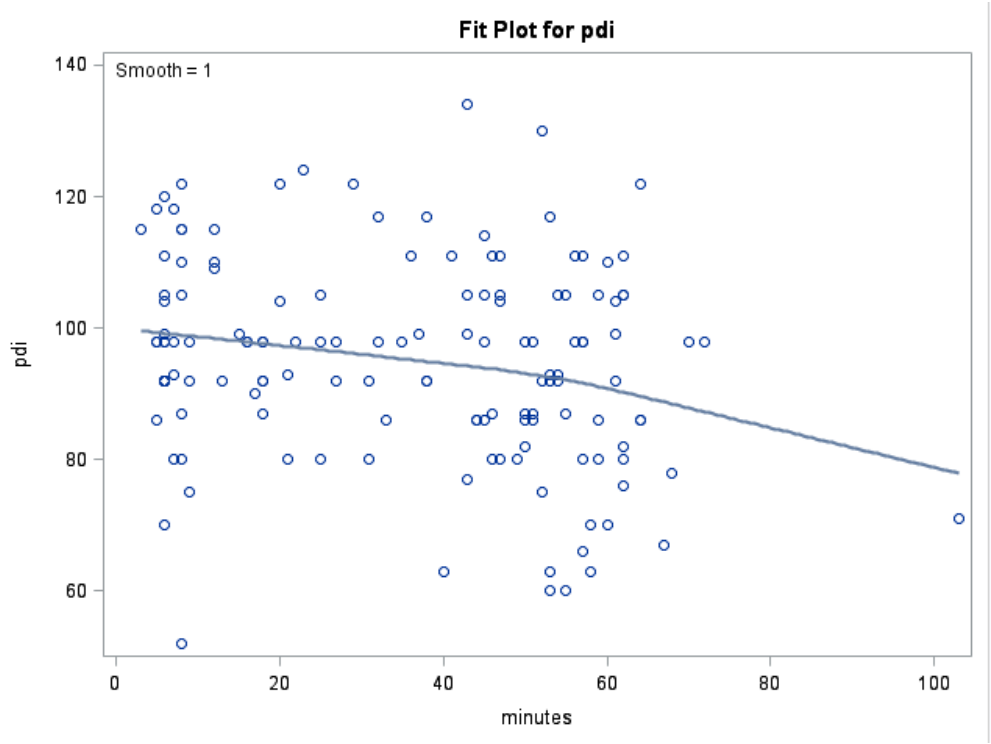
Boxplots

```
proc sort data = dat;
by dhca;
run;
proc boxplot data = dat;
plot minutes * dhca;
run;
```



Scatter Plots and LOESS curves

```
proc gplot data = dat;
plot pdi*minutes;
run;
proc loess data = dat;
model pdi = minutes;
run;
```



Simple Hypothesis Tests

```
# For one-sample T test (H0: mean!=110-->sides=2, mean>110-->sides=u,
  mean<110-->sides=l)
proc ttest data=dat h0=110 sides=2;
var pdi;
run;
# For two-sample T test
proc ttest data=dat sides=2;
class dhca;
var pdi;
run;
# Pearson correlation test
proc corr data=dat;
var pdi minutes;
run;
```

The TTEST Procedure

Variable: pdi

N	Mean	Std Dev	Std Err	Minimum	Maximum
142	95.0986	15.4504	1.2966	52.0000	134.0

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
95.0986	92.5354	97.6618	15.4504	13.8383	17.4909

DF	t Value	Pr > |t|
141	-11.49	<.0001

dhca	N	Mean	Std Dev	Std Err	Minimum	Maximum
0	69	98.4638	13.5934	1.6365	52.0000	124.0
1	73	91.9178	16.4880	1.9298	60.0000	134.0
Diff (1-2)		6.5460	15.1513	2.5439		

dhca	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
0		98.4638	95.1983	101.7	13.5934	11.6432	16.3347
1		91.9178	88.0709	95.7647	16.4880	14.1795	19.7015
Diff (1-2)	Pooled	6.5460	1.5164	11.5755	15.1513	13.5654	17.1604
Diff (1-2)	Satterthwaite	6.5460	1.5428	11.5491			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	140	2.57	0.0111
Satterthwaite	Unequal	137.5	2.59	0.0107

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	72	68	1.47	0.1099

Pearson Correlation Coefficients Prob > |r| under H0: Rho=0 Number of Observations		
	pdi	**minutes**
pdi	1.00000 142	-0.21976 0.0086 142
minutes	-0.21976 0.0086 142	1.00000 171

Linear Regression and Plotting a regression line

```
proc reg data = dat;
model pdi = minutes;
run;
```

The REG Procedure
Model: MODEL1
Dependent Variable: pdi

Number of Observations Read	171
Number of Observations Used	142
Number of Observations with Missing Values	29

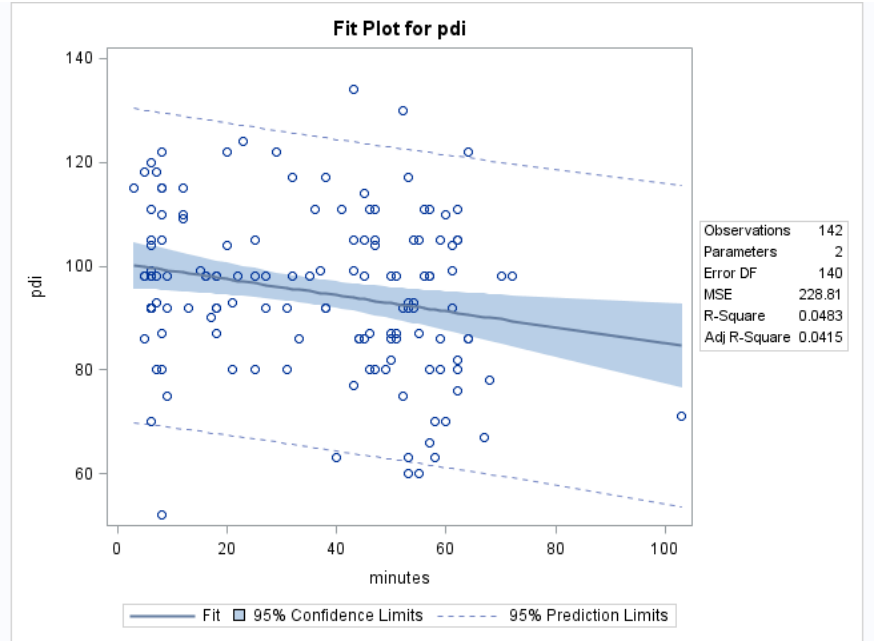
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	1625.47748	1625.47748	7.10	0.0086
Error	140	32033	228.80816		
Corrected Total	141	33659			

Root MSE	15.12641	R-Square	0.0483
Dependent Mean	95.09859	Adj R-Sq	0.0415
Coeff Var	15.90603		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	100.57084	2.41383	41.66	<.0001
minutes	1	-0.15452	0.05797	-2.67	0.0086



Simple Linear Regression: Basic Theory and Application

With Simple Linear Regression, we are looking at models of the form:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

where the Y_i are our outcomes and the X_i are our explanatory variables (covariates).
1. What assumptions about our data and model do we make?

2. Read in 'lab1.csv' with your preferred software, do a one-side T test on X with $H_0 : \mu = 3.5$ and $H_1 : \mu > 3.5$. What is the P-value? Do we reject the null?

3. Fit a regression model, what are the 95% CIs for β_0 and β_1?

4. How to interpret the coefficients

5. What is your prediction of Y if X=5?