

Lecture 14

1. Evaluating a new prediction equation: prediction error criteria
2. GFR example: training and testing subsets
3. Prediction equation does better on the data it was fitted to: optimism
4. Bootstrap estimates of prediction error
5. Bootstrap tests
6. Bootstrap two-sample t -test: brain glucose example
7. Bootstrap one-sample t -test: paired t -test

1

The kidneys filter the blood and remove wastes. The functional unit of the kidney is a **nephron**, and each kidney has about one million nephrons. The central part of the nephron, where the filtration occurs, is called the **glomerulus**. Glomerular filtration rate (**GFR**) is the main measure kidney function.

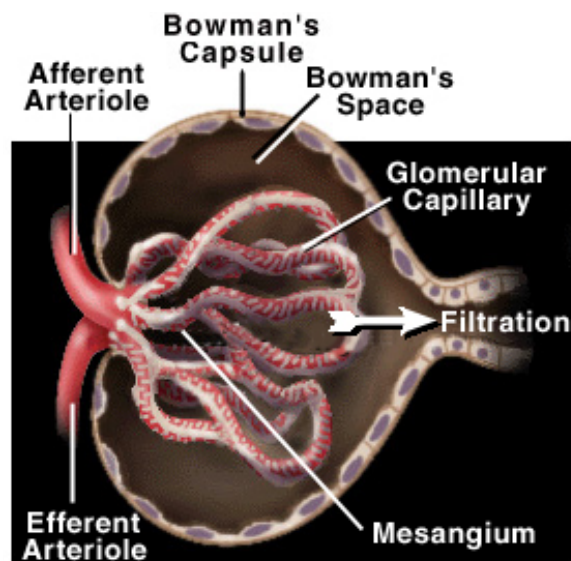


Image from Medical College of Georgia (www.lib.mcg.edu)

2

Evaluating a new prediction equation

The gold standard measurement for GFR is the mean of several determinations by iohexal, which is infused into the blood stream and filtered by the kidneys. Each determination requires several hours of timed blood samples in a clinic, so GFR is rarely determined directly except for research.

GFR is commonly estimated from serum creatinine, measured from a single blood sample, using a published prediction equation that includes age, gender, and race. A new estimate is based on cystatin-C, another chemical that the kidneys filter from blood. In this example, we have directly measured GFR, serum creatinine, and cystatin-C on 140 adults.

Can we show our prediction equation works as well as or better than the published equation?

3

Measures of prediction error, where $\{\hat{y}_i\}$ are predicted values:

- **Bias** = mean of prediction errors = $\sum_i(\hat{y}_i - \text{GFR}_i)/n$
- **Mean Square Prediction Error** = square root of the mean of squared prediction errors = $\{\sum_i(\hat{y}_i - \text{GFR}_i)^2/n\}^{1/2}$
- **Relative Error (accuracy)** = mean absolute value of percent prediction errors
$$\frac{100\%}{n} \sum_i \left| \frac{\hat{y}_i - \text{GFR}_i}{\text{GFR}_i} \right|$$
- **Correlation** between predictions \hat{y}_i and GFR_i

If there is a diagnostic cutoff value for GFR, then we can compute sensitivity and specificity as measures of prediction.

Problem: Is it fair to compare the published equation to our new equation using our data, the same data our equation was fitted to?

4

Training and testing subsets

Randomly divide the data into two subsets:

- **Training:** use this subset to fit the regression equation for predicting
- **Testing:** use this independent subset to assess predictions and compare to other equations

Neither the new equation nor the published equation has an advantage on the testing subset.

```
data two;  
  set ph6470.gfr_reg;  
  train = (ranuni(64256216) > .50 ); Uniform[0,1] random variable  
  subset = "training";  
  if (train=0) then subset="testing";
```

5

Before proceeding, check that the two subsets don't differ on any of the variables in the regression equation:

```
Proc Ttest ci=none data=two; t test for continuous variables  
  class subset;  
  var GFR inv_scr inv_cys age;
```

```
Proc Freq data=two; chi-square for categorical variable  
  tables subset * female /nopercent nocol chisq;
```

6

The TTEST Procedure

Variable	subset	N	Lower CL		Upper CL		Std Dev	Std Err
			Mean	Mean	Mean	Mean		
GFR	testing	68	63.458	71.338	79.219	32.558	3.9482	
GFR	training	72	62.238	69.778	77.317	32.085	3.7813	
GFR	Diff (1-2)		-9.245	1.5605	12.366	32.316	5.4646	
inv_scr	testing	68	0.8033	0.8831	0.9628	0.3294	0.0399	
inv_scr	training	72	0.8311	0.9015	0.9719	0.2996	0.0353	
inv_scr	Diff (1-2)		-0.124	-0.018	0.0867	0.3144	0.0532	
inv_cys	testing	68	0.7755	0.8611	0.9467	0.3537	0.0429	
inv_cys	training	72	0.7858	0.8631	0.9404	0.3289	0.0388	
inv_cys	Diff (1-2)		-0.116	-0.002	0.1121	0.3412	0.0577	
Age	testing	68	43.92	47.735	51.551	15.764	1.9116	
Age	training	72	45.051	49.042	53.032	16.983	2.0014	
Age	Diff (1-2)		-6.791	-1.306	4.1779	16.402	2.7736	

Variable	Method	Variances	DF	t Value	Pr > t
GFR	Pooled	Equal	138	0.29	0.7756
GFR	Satterthwaite	Unequal	137	0.29	0.7757
inv_scr	Pooled	Equal	138	-0.35	0.7288
inv_scr	Satterthwaite	Unequal	135	-0.35	0.7295
inv_cys	Pooled	Equal	138	-0.03	0.9729
inv_cys	Satterthwaite	Unequal	136	-0.03	0.9730
Age	Pooled	Equal	138	-0.47	0.6384

7

subset	female		
Frequency	0	1	Total
testing	34	34	68
	50.00	50.00	
training	34	38	72
	47.22	52.78	
Total	68	72	140

Statistics for Table of subset by female

Statistic	DF	Value	Prob
Chi-Square	1	0.1080	0.7424

Are these subsets OK or do we need to select new ones?

Fit the regression equation using the training subset ($n = 72$):

Parameter	Estimate	Standard Error	t Value	Pr > t
Intercept	10.81827259 B	9.95245507	1.09	0.2809
inv_scr	21.80002096	11.00383743	1.98	0.0517
inv_cys	61.59686164	10.21099476	6.03	<.0001
Age	-0.35749483	0.11253482	-3.18	0.0023
female 0	7.78284108 B	3.49815661	2.22	0.0295
female 1	0.00000000 B	.	.	.

How do we apply this to the testing subset to get predicted values?

9

```
data pred;
  set two; contains both training and testing subsets
  if (train=1) then pred_GFR = GFR;
  else pred_GFR = . ;

Proc GLM data=pred;
  class female;
  model pred_GFR = inv_scr inv_cys age female / solution;
  output out=three p = yhat;
```

How does this fit the model on the training subset only?

Will it fit predicted values \hat{y}_i for both subsets?

How do we get prediction errors ($\hat{y}_i - \text{GFR}_i$)?

Now calculate the prediction error criteria—*one observation for each subset*:

```
proc sort data=three; output data from Proc GLM
  by train;

data four;
  set three;
  by train; creates first.train and last.train
  if (first.train=1) then do; zero out at start of subset
    bias=0; RMSE=0; accuracy=0; sample_size=0;
    end; why count sample size?
  retain bias RMSE accuracy sample_size;
  resid = yhat - GFR; reverse of usual
  bias = sum(bias, resid);
  RMSE = sum(resid**2, RMSE);
  temp = 100.0 * abs(resid)/GFR;
  accuracy = sum(temp, accuracy);
  if (resid NE .) then sample_size = sample_size + 1;
```

11

```
if (last.train=1) then do; at the end of a subset, find means
  bias = bias/sample_size;
  RMSE = sqrt(RMSE/sample_size);
  accuracy = accuracy/sample_size;
  output; output one observation for each subset
end;
```

12

Obs	subset	bias	RMSE	accuracy	sample_size
1	testing	-1.40076	15.8454	21.0944	68
2	training	-0.00000	13.6586	19.7762	72

Report the results from the testing subset. These are the error criteria for comparison with the published equation.

Bias (mean of residuals) is always zero in the training subset because in regression with an intercept, residuals sum to zero.

RMSE and relative error (accuracy) are *larger in the testing subset*.

This is usually so: a prediction equation works better on the data it was fitted to.

Prediction error criterion from the training sample underestimates prediction error on the testing sample.

13

Find the difference between the prediction error criteria on the testing and training subsets:

this is **optimism**, the amount that prediction error on the training sample underestimates prediction error on a new sample.

```
proc sort data=four; by subset;
data five;
  set four;
  if (subset="testing") then do;
    bias0 = bias;  RMSE0 = RMSE;  accuracy0= accuracy;
  end;
  retain bias0 RMSE0 accuracy0 ;
  if (subset="training") then do;
    d_bias = bias - bias0;  d_RMSE = RMSE - RMSE0;
    d_accuracy = accuracy - accuracy0;
  output;
  end;
```

14

Obs	subset	bias	RMSE	accuracy	sample_size
1	testing	-1.40076	15.8454	21.0944	68
2	training	-0.00000	13.6586	19.7762	72

Obs	d_bias	d_RMSE	d_accuracy
1	1.40076	-2.18687	-1.31824

These differences are estimates of **optimism** from one training subset.

Prediction error criterion from training subset + optimism

= prediction error criterion from testing subset.

15

Bootstrap approach

Idea: Use the whole sample to fit a prediction equation and get prediction error. Bootstrap to get replicate estimates of optimism. Then add the bootstrap optimism to the prediction error from the original sample.

1. Draw B samples $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \dots, \mathbf{x}^{*B}$ with replacement from the data \mathbf{x}
2. Fit the regression model to each bootstrap sample \mathbf{x}^{*b} : treat each bootstrap sample as the training sample and the original data \mathbf{x} as the testing sample. Compute optimism^{*b} = prediction error (\mathbf{x} , testing) – prediction error (\mathbf{x}^{*b} , training).
3. Average the bootstrap replicates {optimism^{*b}}. Add these to the prediction errors from the original data \mathbf{x} .

16

We need to repeat the earlier SAS program for testing and training subsets using the bootstrap samples. Unfortunately, the %boot macro draws all the bootstrap samples before calling %analyze, so it doesn't work to put all the earlier code inside %analyze.

My awkward SAS program:

1. Call %boot to draw bootstrap samples (bootdata) with a dummy %analyze macro
2. Make a copy of the original data with an indicator flag = 1
3. For $b = 1, \dots, B$, stack the flagged data on top of \mathbf{x}^{*b} , and set the response to missing for \mathbf{x} , where flag = 1.

Run Proc GLM, get fitted values, compute prediction errors r_b .

This gives B datasets, each with 2 observations: $r_b(\mathbf{x}^{*b})$ and $r_b(\mathbf{x})$.

To run Proc GLM inside a DO-loop, we will need to write a macro.

17

4. For $b = 1, \dots, B$, stack up (set) the prediction error datasets.

This is a separate DO-loop, inside a SET statement.

5. In the combined dataset, compute: optimism^{*b} = prediction error $r_b(\mathbf{x})$ – prediction error $r_b(\mathbf{x}^{*b})$
6. Average the bootstrap replicates {optimism^{*b}}. Add these to the prediction errors from the original data \mathbf{x} .

Bootstrapping prediction error (optimism)

Step 1. Draw bootstrap samples.

```
%macro analyze(data=,out=); dummy: the correlations are not of interest
proc corr noprint data=&data outs=&out;
  var inv_scr inv_cys;
  %bystmt;
run;
data &out;
  set &out;
  if ( _type_='CORR' and _name_='inv_scr' );
  corr =inv_cys;
  drop _TYPE_ _NAME_ inv_scr inv_cys;
  run;
%mend analyze;

%boot(data=ph6470.gfr_reg,samples = 200, biascorr=0,chart=0);

draws 200 bootstrap samples, stacked in "bootdata"
```

19

Step 2. Copy original data with an indicator flag = 1

```
data flag;
  set ph6470.gfr_reg;
  flag = 1; * flag=1 for original sample, 0 for bootstrap;
```

Step 3. For $b = 1, \dots, B$, stack the flagged data on top of \mathbf{x}^{*b} , run Proc GLM, get fitted values, compute prediction errors r_b .

```
%macro optimism (data=,B=);
%do i=1 %to &B; macro DO-loop: counter i is a macro-variable &i
  data strap&i; select bootstrap sample i
  set &data;
  if _sample_=&i;
data pred&i; one input dataset for each bootstrap sample i
  set strap&i work.flag;
  _sample_ =&i;
  if (flag=1) then pred_GFR = .;
  else do; pred_GFR = GFR ; flag=0; end;
```

20

```

Proc GLM data=pred&i noprint;
  model pred_GFR = inv_scr inv_cys age female;
  output out=temp&i p = yhat; one output dataset for each bootstrap sample i
run;
proc sort data=temp&i;
  by flag;
run;
data temp&i; sum the prediction errors across observations
  set temp&i;
  by flag;
  if (first.flag=1) then do;
    bias=0; RMSE=0; accuracy=0; sample_size=0;
    end;
  retain bias RMSE accuracy sample_size;
  resid = yhat - GFR;
  if (resid NE .) then sample_size = sample_size + 1;

```

21

```

  bias = sum(bias, resid);
  RMSE = sum(resid**2, RMSE);
  tmp = 100.0 * abs(resid)/GFR;
  accuracy = sum(tmp, accuracy);
  if (last.flag=1) then do;
    bias = bias/sample_size;
    RMSE = sqrt(RMSE/sample_size);
    accuracy = accuracy/sample_size;
    output; output 2 observations: flag=0 and flag=1
    end;
  keep _sample_ flag bias RMSE accuracy sample_size;

%end; end of first DO-loop

```

22

```

data boot_replicates; Step 4: stack prediction errors
  set   %do i=1 %to &B;
        temp&i
        %end ;
%mend optimism ;

%optimism(data=work.bootdata, B=200); call the macro above

proc print data=boot_replicates(obs=20);

```

23

Obs	_sample_	flag	bias	RMSE	accuracy	sample_size
1	1	0	0.00000	12.8224	17.5395	140
2	1	1	1.13874	14.9659	21.1608	140
3	2	0	0.00000	15.0381	21.8281	140
4	2	1	-0.41592	14.7635	20.4765	140
5	3	0	0.00000	13.3444	19.8810	140
6	3	1	1.33701	14.6983	21.1171	140
7	4	0	-0.00000	13.7881	19.4232	140
8	4	1	0.14610	14.8936	20.4979	140
9	5	0	0.00000	13.4803	17.5611	140
10	5	1	1.73094	14.7966	20.9748	140
11	6	0	0.00000	14.8362	18.7613	140
12	6	1	3.34088	15.1832	22.1807	140
13	7	0	-0.00000	13.6539	16.8368	140
14	7	1	1.00582	14.7241	20.6480	140
15	8	0	0.00000	14.7710	19.3176	140
16	8	1	1.71886	14.8440	21.1802	140
17	9	0	0.00000	14.1820	19.1077	140
18	9	1	0.56883	14.7337	20.6924	140
19	10	0	0.00000	14.0154	18.1580	140
20	10	1	0.77697	15.0521	20.9821	140

24

Step 5. Compute optimism^{*b}

```
proc sort data=boot_replicates;
  by _sample_ flag;

data optimism_replicates;
  set boot_replicates;
  by _sample_ flag;
  if (flag=0) then do;
    bias0 = bias; RMSE0 = RMSE; accuracy0= accuracy;
  end;
  retain bias0 RMSE0 accuracy0 ;
  if (flag=1) then do;
    d_bias = bias - bias0; d_RMSE = RMSE - RMSE0;
    d_accuracy = accuracy - accuracy0;
    output;
  end;
  keep _sample_ d_bias d_RMSE d_accuracy;
```

25

First 20 optimism replicates:

Obs	_sample_	d_bias	d_RMSE	d_accurac
1	1	1.13874	2.14346	3.62126
2	2	-0.41592	-0.27459	-1.35166
3	3	1.33701	1.35391	1.23613
4	4	0.14610	1.10547	1.07471
5	5	1.73094	1.31627	3.41373
6	6	3.34088	0.34707	3.41944
7	7	1.00582	1.07024	3.81118
8	8	1.71886	0.07303	1.86259
9	9	0.56883	0.55169	1.58467
10	10	0.77697	1.03664	2.82402
11	11	-1.25235	-0.35429	-1.90702
12	12	-0.77213	0.05118	-0.55257
13	13	-2.05813	-0.14681	-0.03227
14	14	1.00043	1.17405	2.12095
15	15	-0.70870	0.52491	0.43950
16	16	0.81215	0.42442	1.32588
17	17	-0.40309	0.71109	1.83802
18	18	-1.47694	-1.79046	-5.48530
19	19	-0.45316	1.79518	2.01903
20	20	-0.89760	1.07245	1.49493

26

Step 6. Average bootstrap replicates {optimism^{*b}}.

```
proc means n mean stddev data=optimism_replicates; estimates of optimism  
var d_bias d_RMSE d_accuracy;
```

Compute prediction error from original sample

```
Proc GLM data=ph6470.gfr_reg;  
model GFR = inv_scr inv_cys age female;  
output out=z1 r = resid;
```

```
data z2;  
set z1;  
accuracy = 100.0*abs(resid)/GFR;
```

```
proc means n mean stddev data=z2;  
var accuracy;
```

27

The MEANS Procedure

Variable	N	Mean	Std Dev
d_bias	200	-0.0124186	1.2197518
d_RMSE	200	0.6966326	1.1271686
d_accuracy	200	0.7822725	2.0808914

The GLM Procedure

Dependent Variable: GFR

R-Square	Coeff Var	Root MSE	GFR Mean
0.792728	21.09479	14.87936	70.53571

Analysis Variable : accuracy

N	Mean	Std Dev
140	20.6858002	25.6365356

28

Bootstrap prediction errors

Bias: $0 - 0.0124186 = -0.01$

RMSE: $14.87936 + 0.6966326 = 15.6$

Relative error (%): $20.6858002 + 0.7822725 = 21.5\%$

Units for bias and RMSE are the same as for the response GFR.

29

Bootstrap tests

So far, we have used the bootstrap methods this way:

1. draw B samples with replacement from the data,
2. calculate the statistic $\hat{\theta}_b^*$ from each bootstrap sample, $b = 1, \dots, B$,
3. use the histogram of the bootstrap statistics $\{\hat{\theta}^*(b)\}$ to find confidence intervals and standard errors.

Hypothesis testing differs from confidence intervals and standard errors: tests have a null hypothesis, and the p -value for the test is calculated assuming this null hypothesis.

To apply the bootstrap, we need to draw the bootstrap samples from the *null hypothesis distribution*.

30

The two sample t -test compares two population means μ_Y and μ_Z by comparing estimates of these means from independent samples $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ from population Y and $\mathbf{z} = \{z_1, z_2, \dots, z_m\}$ from population Z . The standard assumptions are:

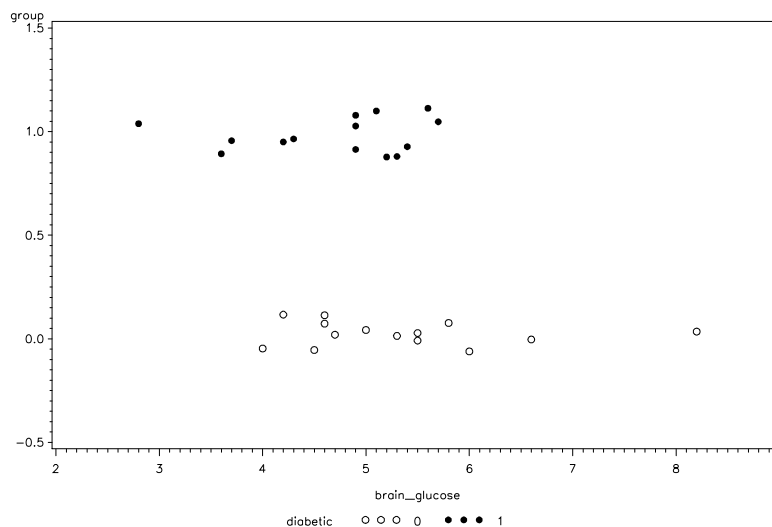
1. The data are simple random samples
2. The two populations are Normal
3. The two populations have the same variance
4. The observations are correctly labeled with their population:
no misclassification.

The null hypothesis is: $\mu_Y = \mu_Z$.

The samples do not have equal averages. How do we sample from the data under this *null hypothesis distribution*?

Brain glucose example

Magnetic resonance imaging gives researchers a non-invasive method to measure chemicals in the brain minute by minute. One study examined levels of blood sugar (glucose) in the brains of 14 people with diabetes and 14 healthy people.



The TTEST Procedure

Variable	diabetic	Lower CL		Upper CL		Std Dev	Std Err
		N	Mean	Mean	Mean		
brain_		14	4.6832	5.3214	5.9596	1.1054	0.2954
glucose	0						
brain_		14	4.1943	4.6857	5.1771	0.8511	0.2275
glucose	1						
brain_	Diff (1-2)		-0.131	0.6357	1.4021	0.9865	0.3728
glucose							

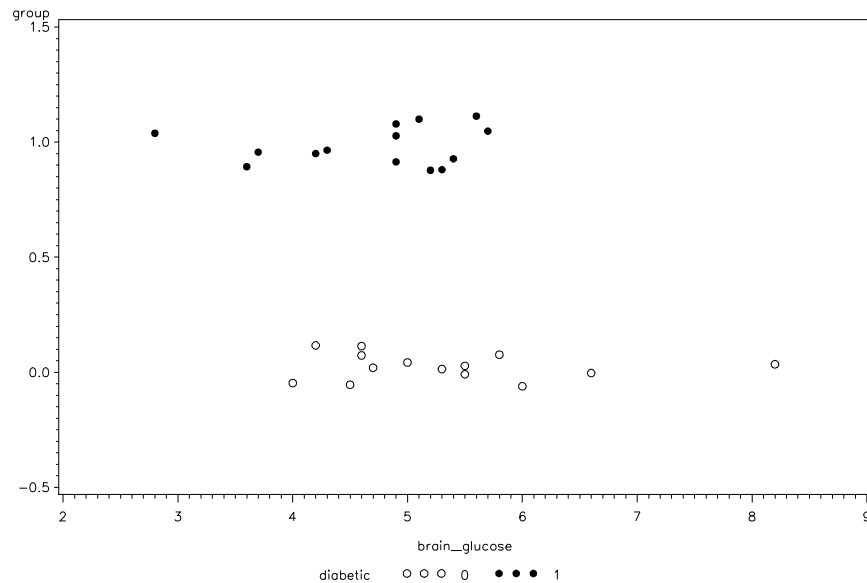
Variable	Method	Variances	DF	t Value	Pr > t
brain_glucose	Pooled	Equal	26	1.71	0.1001
brain_glucose	Satterthwaite	Unequal	24.4	1.71	0.1009

According to the SAS Help, the test statistic calculated for **Variances Unequal** is

$$t = \frac{\bar{y} - \bar{z}}{\sqrt{SD(\mathbf{y})^2/n + SD(\mathbf{z})^2/m}} = \frac{\bar{y} - \bar{z}}{\sqrt{SE(\bar{y})^2 + SE(\bar{z})^2}}$$

but both observed test statistics are the same, so $t_{obs} = 1.71$.

The test indicates no difference. But these are small samples that looked skewed in opposite directions—perhaps the t -test is missing something?



We can use the bootstrap to look at the sampling distribution of t .

Bootstrap 2-sample t-test

We have two independent samples: $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ from population Y and $\mathbf{z} = \{z_1, z_2, \dots, z_m\}$ from population Z .

1. Calculate the *observed* test statistic from the samples:

$$t_{obs} = \frac{\bar{y} - \bar{z}}{\sqrt{\text{SE}(\bar{y})^2 + \text{SE}(\bar{z})^2}}$$

This version of the t-statistic does not assume equal population variances.

2. Create two transformed data sets \mathbf{y}' and \mathbf{z}' with equal means to satisfy the null hypothesis.

Although we want equal means, we don't want to change the standard deviations. How can we do that?

35

Let \bar{y} be the mean of sample \mathbf{y} , and \bar{z} be the mean of sample \mathbf{z} .

Subtract \bar{y} from each observation in sample \mathbf{y} :

$$\mathbf{y}' = \{y'_1, y'_2, \dots, y'_n\} = \{(y_1 - \bar{y}), (y_2 - \bar{y}), \dots, (y_n - \bar{y})\}$$

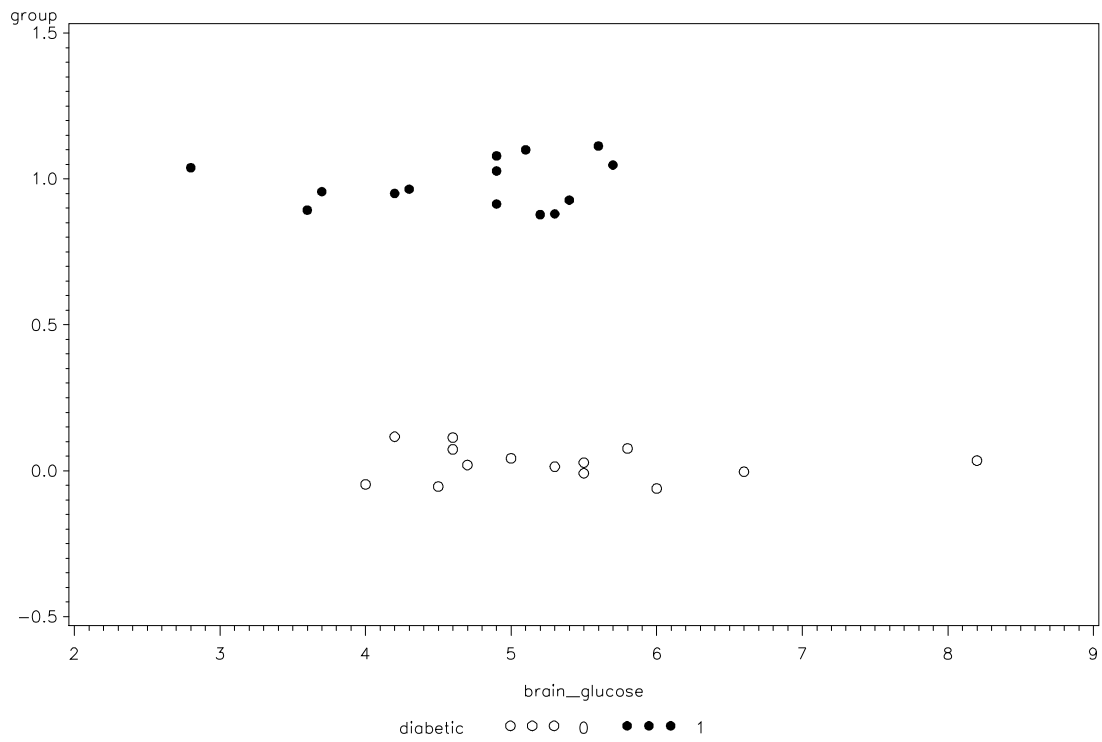
Subtract \bar{z} from each observation in sample \mathbf{z} :

$$\mathbf{z}' = \{z'_1, z'_2, \dots, z'_m\} = \{(z_1 - \bar{z}), (z_2 - \bar{z}), \dots, (z_m - \bar{z})\}$$

Both of these transformed data sets have mean zero, fulfilling the null hypothesis.

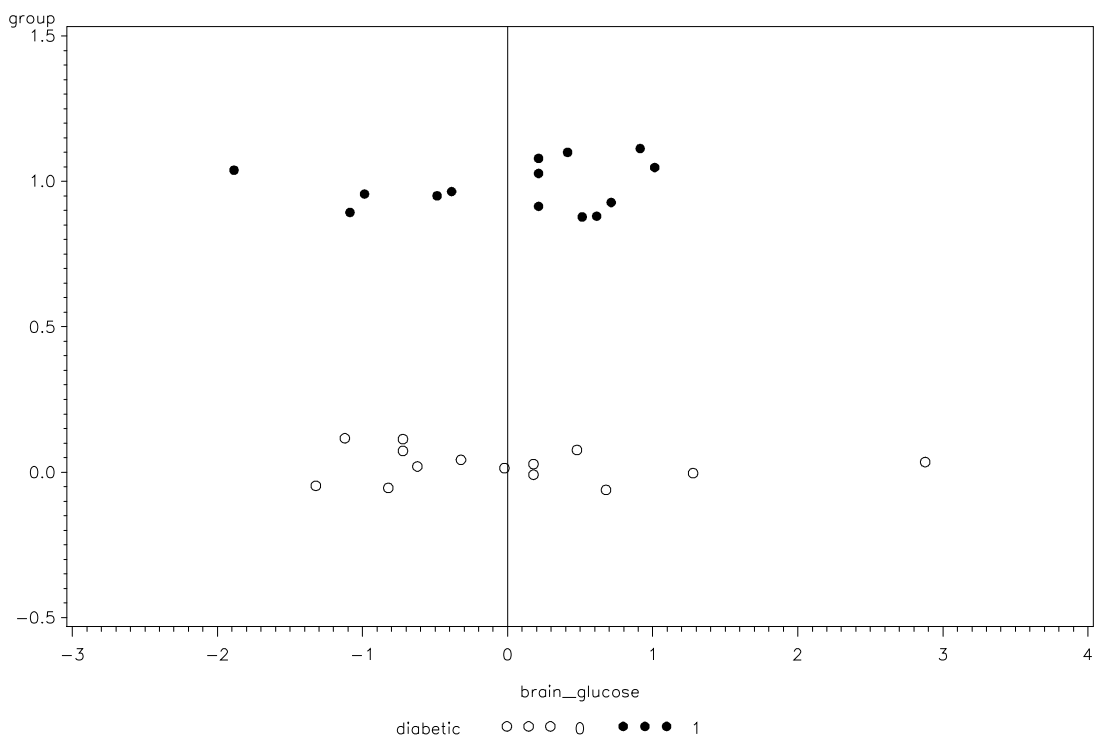
$$\frac{1}{n} \sum_i y'_i = \frac{1}{n} \sum_i (y_i - \bar{y}) = \frac{1}{n} \left(\sum_i y_i \right) - \frac{1}{n} (n\bar{y}) = \bar{y} - \bar{y} = 0$$

36



Original data: samples y and z .

37



Shifted data: samples y' and z' .

38

Because we need to bootstrap each sample separately, we must make two data sets and then run the bootstrap macro on each one. We make the shift while making separate data sets.

From Proc Ttest, we have $\bar{y} = 5.3214$ and $\bar{z} = 4.6857$.

```
data control diabetes ; Make data sets for each group with zero mean
  set pubh.brain_glucose;
  if (diabetic=0) then do;
    bg = brain_glucose - 5.3214;
    output control ;
  end;
  if (diabetic=1) then do;
    bg = brain_glucose - 4.6857;
    output diabetes ;
  end;
```

39

3. Write %analyze to calculate the mean and standard error of a bootstrap sample.

```
proc means noprint data=pubh.brain_glucose; regular code to test the idea
  var brain_glucose;
  output out=a mean=xbar stderr=se;
proc print data=a;
```

Obs	_TYPE_	_FREQ_	xbar	se
1	0	28	5.00357	0.19290

```
%macro analyze(data=,out=);
  proc means noprint data=&data;
    var bg;
    output out=&out mean=xbar stderr=se;
    %bystmt;
  run;
  data &out;
  set &out;
  drop _type_ _freq_ ;
  run;
%mend analyze;
```

40

4. Perform *two* parallel bootstraps of size B : one from \mathbf{y}' and one from \mathbf{z}' .

This means two calls of the bootstrap macro, which produces two `bootdist` datasets of means and SEs.

```
%boot (data=control , samples=1000, chart=0, biascorr=0)
```

```
data boot_control;
```

```
set bootdist; rename the first data set in order to save it
```

```
%boot (data=diabetes , samples=1000, chart=0, biascorr=0)
```

If you don't rename `bootdist` , the second call to `%boot` will overwrite it.

41

5. Merge the two `bootdist` datasets by bootstrap sample number. Calculate the test statistic

$$t_b^* = \frac{\bar{y}^* - \bar{z}^*}{\sqrt{\text{SE}(\bar{y}^*)^2 + \text{SE}(\bar{z}^*)^2}} \quad \text{for } b = 1, \dots, B,$$

using the means and standard errors from the bootstrap samples.

6. In the same dataset calculate an indicator for $|t_b^*| \geq |t_{obs}|$.

Use Proc Freq to find the **bootstrap approximate two-sided p -value** :

$$\frac{1}{B} \{\text{Number of } |t_b^*| \geq |t_{obs}|\}.$$

This is the proportion of test statistics from the H_0 distribution that are more extreme than the one we observed.

42

```

data tstat;
  merge boot_control (rename = (xbar = mean_c se = se_c))
        bootdist (rename = (xbar = mean_d se = se_d));

  tstat = (mean_c - mean_d)/sqrt(se_c*se_c + se_d*se_d);

  observed = 1.71;
  bigger = (abs(tstat) GE observed);

```

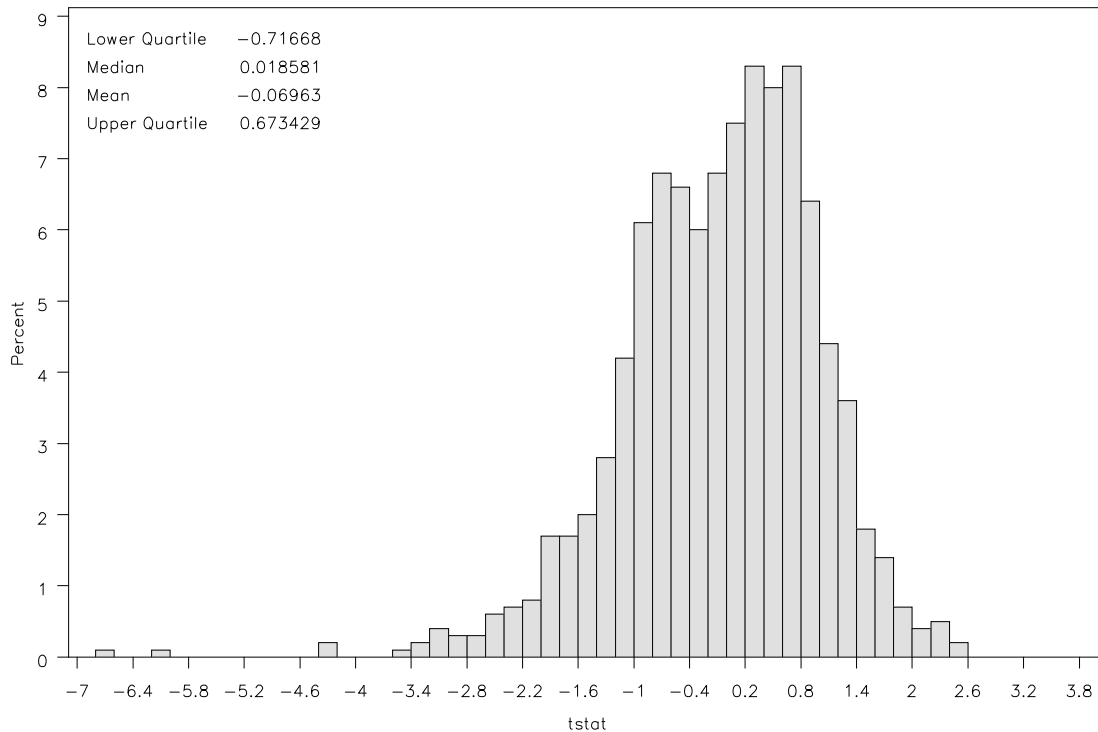
bigger=1 when $|t_b^*| \geq |t_{obs}| = 1.71$.

43

Here are the first 15 rows of the data tstat

Obs	_sample_	mean_c	se_c	mean_d	se_d	tstat	observed	bigger
1	1	0.10003	0.37712	0.14287	0.18380	-0.10212	1.71	0
2	2	-0.09997	0.19950	-0.08570	0.16871	-0.05462	1.71	0
3	3	0.13574	0.35990	0.29287	0.13795	-0.40767	1.71	0
4	4	0.12146	0.35514	0.10716	0.19540	0.03528	1.71	0
5	5	0.42146	0.33704	0.04287	0.18230	0.98800	1.71	0
6	6	-0.39997	0.18932	0.17144	0.17904	-2.19288	1.71	1
7	7	0.42860	0.43066	0.22859	0.25888	0.39806	1.71	0
8	8	0.56431	0.30416	-0.09284	0.20767	1.78433	1.71	1
9	9	0.10003	0.26503	-0.40713	0.26503	1.35312	1.71	0
10	10	0.10003	0.30816	0.22859	0.16996	-0.36530	1.71	0
11	11	-0.11426	0.32386	0.17144	0.25043	-0.69786	1.71	0
12	12	-0.19997	0.20944	0.15716	0.18148	-1.28866	1.71	0
13	13	-0.05711	0.33292	0.24287	0.25342	-0.71698	1.71	0
14	14	-0.37854	0.15253	0.08573	0.17834	-1.97839	1.71	1
15	15	0.40717	0.35401	0.02144	0.22519	0.91936	1.71	0

44



Histogram of t-statistics drawn under $H_0: \mu_{\text{Control}} = \mu_{\text{Diabetic}}$.

45

Use Proc Freq to find the bootstrap approximate two-sided p -value:

$$\frac{1}{B} \{\text{Number of } |t_b^*| \geq |t_{\text{obs}}|\}.$$

```
Proc Freq data=tstat;
  table bigger;
```

bigger	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	915	91.50	915	91.50
1	85	8.50	1000	100.00

The bootstrap 2-sided p -value = $85/1000 = 0.085$. The t -test gave $p = 0.1009$.

In this small sample, with normality in doubt, the bootstrap provides reassurance that the t -test is not missing a real difference.

46

One-sample t-test: paired data

After the first analysis, it came out that participants in the brain glucose fMRI study were paired: diabetic/control.

Experimental data should be analyzed according to experimental design, so the appropriate test is based on differences within pairs.

Here is a paired t -test of the within-pair differences d_i in brain glucose:

```
Proc Means  n mean stddev t probt  data=pubh.brain_paired;  
var d_brain_glucose;
```

```
Analysis Variable : d_brain_glucose  
N           Mean           Std Dev      t Value     Pr > |t|  
14          0.6357143        1.5906492    1.50         0.1587
```

47

The usual null hypothesis of the paired t -test is that the population-mean of the differences is zero.

To make a sample that satisfies this null hypothesis, we subtract the mean difference $\bar{d} = 0.6357143$ from each observation in the sample.

```
data centered;  
set pubh.brain_paired;  
centered_diff = d_brain_glucose - 0.6357143;
```

We will draw bootstrap samples from the data set `centered`.

This is like the one-sample problems we have seen (correlation, kappa) except that we'll get a p -value instead of a confidence interval from the bootstrap distribution.

48

```

%macro analyze(data=,out=);
  proc means noprint data=&data;
    var centered_diff;
    output out=&out t=t_stat;
    %bystmt;
  run;
  data &out; remove extra variables
    set &out;
    drop _type_ _freq_ ;
  run;
%mend analyze;

%boot(data=centered,samples=1000,chart=0, biascorr=0)

data as1;
  set bootdist;
  bigger2 = (abs(t_stat) GE 1.50 ); two-sided p-value
  bigger1 = (t_stat GE 1.50 ); one-sided p-value, using right side

proc freq data=as1;
  tables bigger1 bigger2;

```

We use Proc Freq to count the number of extreme t -statistics.

49

one-sided

bigger1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	952	95.20	952	95.20
1	48	4.80	1000	100.00

two-sided

bigger2	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	829	82.90	829	82.90
1	171	17.10	1000	100.00

What are the two p -values? Why is twice the one-sided not equal to the two-sided?

50

