Discussion 5

R operations

1. Density Plot

   The following examples show examples of density plots for the female sockeye salmon mass data set. First, read in the data. There is a single variable named mass.

   The density plot draws a curve which is estimated by averaging many (50 by default) histograms, with the breakpoints shifted slightly for each. To suppress the plotting of points (useful when the sample size is enormous), add the argument $plot.points = F$.

```
> plot(densityplot(~mass, data = salmon, xlab = "Body Mass (kg)",
+ylab = "Density", n = 201))
```

2. Graphs to Compare Distributions

   The following graph compares the distribution of the number of broods for each group with histograms. The vertical bar in the formula species the variable on the left to be split according to the variable on the right and displayed in dierent panels. The layout argument species one column and two rows (so the histograms are aligned vertically, making them easier to compare).

```
> plot(histogram(~broods | mating, data = pscorp, layout = c(1, 2),
+ breaks = seq(-0.5, 7.5)))
```

3. One Sample T test and Confidence Interval

   We demonstrate using $t.test()$ for one-sample condence intervals and hypothesis tests using a sample of 25 body temperatures.

```
> temp = read.table("temperature.txt", header = T)
> str(temp)
'data.frame':
25 obs. of 1 variable:
 temperature: num 98.4 98.6 97.8 98.8 97.9 99 98.2 98.8 98.8 99 ...
```

   The variable can be specied from the data frame with the $ operator. The mean of the null hypothesis is set with mu=98.6 and the condence level is set with $conf.level = 0.99$ (this can be shortened to $conf$.

```
> t.test(temp$temperature, mu = 98.6, conf = 0.99)
One Sample t-test
data: temp$temperature
t = -0.5606, df = 24, p-value = 0.5802
alternative hypothesis: true mean is not equal to 98.6
99 percent confidence interval:
98.14485 98.90315
sample estimates:
mean of x
98.52
```

4. Two Sample T test

   The base function t.test() can be used for both paired and independent sample t-tests and con dence intervals. See slide notes on usage for arguments such as: $paired = T$, $alternative = greater$,$conf = 0.99$, and $var.equal = T$. Compare the previous result to the two-sample independent t-test.

```
> sample1 = with(pscorp, broods[mating == "Same"])
> sample2 = with(pscorp, broods[mating == "Different"])
> t.test(sample1, sample2, alternative = "less")

Welch Two Sample t-test
data: sample1 and sample2
t = -2.3424, df = 28.883, p-value = 0.01313
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
-Inf -0.3911856
sample estimates:
mean of x mean of y
2.200
3.625
```

5. The Bootstrap

   We used the bootstrap to nd a 95% condence interval for the salmon data in lecture. Here is the R code for this. The basic idea is to create a large matrix with B rows and n columns where n is the sample size of the original data and B is the number of bootstrap data sets we wish to replicate. We use matrix() to create the matrix and *sample*() with $replace = T$ to sample data with replacement. The function *apply*() with second argument 1 (the number one) and third argument mean applies the function *mean*() to each row of the matrix. Finally, we use *quantile*() to nd the corresponding quantiles of the sample. Here is an application of the bootstrap using $B = 10,000$.

   You can also use *boot.ci*() in *boot.R* returns a list with a confidence interval and the entire bootstrap sample of the test statistic. See the sockeye salmon mass example in the homework assignment.

```
> B = 10000
> n = length(salmon$mass)
> mass.boot = apply(matrix(sample(salmon$mass, size = n * B, replace = T),
> nrow = B, ncol = n), 1, mean)
> quantile(mass.boot, c(0.025, 0.975))
2.5% 97.5%
1.959254 2.100352

> mass.boot.2 = apply(matrix(sample(salmon$mass, size = n * B, replace = T),
+ nrow = B, ncol = n), 1, mean)
> quantile(mass.boot.2, c(0.025, 0.975))
2.5%   97.5%
1.960877 2.101009
```

   Compare to the t-test.

```
> t.test(salmon$mass)
One Sample t-test
data: salmon$mass
t = 56.8042, df = 227, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
1.957804 2.098512
sample estimates:
mean of x
2.028158
```

The 95% condence intervals are identical when rounded to two decimal places.

6. Permutation Test/ Randomization Test

Randomization test to test if two population means were equal. Here is R code to read in the data set using read.csv() which expects a text le where the rst line is a header line with variable names and subsequent lines have data. Values on each line are separated by commas.

You can also use *perm.test*() in *boot.R* returns a list with a p-value and the entire permutation distribution of the test statistic. See the pseudoscorpion example in the homework assignment.

```
> pscorp = read.csv("pseudoscorpions.csv")
> str(pscorp)
'data.frame':
36 obs. of 2 variables:
$ mating: Factor w/ 2 levels "Different","Same": 2 2 2 2 2 2 2 2 2 2 ...
$ broods: int 4 0 3 1 2 3 4 2 4 2 ...
```

To carry out the randomization test, we will write a special function that will compute the mean of the rst 20 observations, the mean of the next 16 observations, and return the dierence.

```
> f = function(x) {
+
return(mean(x[1:20]) - mean(x[21:36]))
+}
```

Next, we create an array to store the dierence in means for each randomized data set. We will do this R = 100, 000 times. We will the array with missing values (NA) which we will replace.

```
> R = 100000
> out = rep(NA, R)
```

The function sample() with only one argument consisting of an array returns a random permutation of the elements of the array. We think of this as the rst 20 elements being the one randomly sampled group and the next 16 as the second group. This command rerandomizes one time and calls f() to nd the dierence in means.

```
> f(sample(pscorp$broods))
[1] 0.0375
```

Now, we ask R to do this R = 100, 000 times with the for() command. The variable i is set to each value from 1 to R, and the dierence in means for that particular rerandomization is stored in out[i]. The test statistic is found by applying f() to the original data (which works because the data is ordered with the Same group having the rst 20 observations and the Dierent group having the nal 16 observations). The p-value is the proportion of randomized dierences in sample means that are less than this test statistic.

```
> test.stat = f(pscorp$broods)
> for (i in 1:R) {
+ out[i] = f(sample(pscorp$broods))
+ }
> p.value = sum((out <= test.stat))/R
> p.value
[1] 0.01515
```