Week 11 Quiz - Hypothesis Testing Part 2 - Solutions

FIT5197 teaching team

Note you will need to use the z- and t- tables in the unit [Formula Sheet (https://lms.monash.edu/mod/resource/view.php?id=7439150)](https://lms.monash.edu/mod/resource/view.php?id=7439150) to answer the following questions.

Question 1

Take-Home Pay. Who earns more: Married or unmarried people? A sample is drawn from the population to find out. The sample mean weekly income for married and unmarried people is found to be \$639.60 and \$759.20, respectively. The population standard deviations of weekly income for married and unmarried people are known to be \$60 and \$90, respectively. The number of samples used to generate the sample means for the married and unmarried people are $n = 40$ and $m = 60$, respectively. Perform a hypothesis test to see if married and unmarried people earn the same income.

Answer 1

So the population standard deviations are known and if we assume the married and unmarried income populations both follow a Gaussian distribution, we can use the two-sample z-test. Looking up the following two-sample test table in the formula sheet we can see why we would choose the z-statistic and the table gives you the exact formulation of the two-sample z-statistic to use:

assume dataset of count n with mean $\bar{X}$ and sample variance S^2 and a second dataset of count m with mean $\bar{Y}$ and sample variance T^2:

assumptions	null-hypo.	test statistic
Gaussian, σ_1^2, σ_2^2 known	$\Delta\mu_0$	$Z = \frac{\bar{X}-\bar{Y}-\Delta\mu_0}{\sqrt{\sigma_1^2/n+\sigma_2^2/m}}$
Gaussian, $\sigma_1^2 = \sigma_2^2$ unknown but equal	$\Delta\mu_0$	$t_{n+m-2} = \frac{\bar{X}-\bar{Y}-\Delta\mu_0}{S_P\sqrt{\frac{1}{n}+\frac{1}{m}}}$ for $S_P^2 = \frac{(n-1)S^2+(m-1)T^2}{n+m-2}$
Gaussian, $\sigma_1^2 \neq \sigma_2^2$ unknown, using CLT	$\Delta\mu_0$	use 1st case for $\sigma_1^2 = S^2, \sigma_2^2 = T^2$, assuming n, m are large

Now before computing the z-statistic we need to define our hypotheses. We want to test if married and unmarried people earn the same income so our null hypothesis can be that the population mean income of married people, μ_1, and the population mean income of unmarried people, μ_2, are equal. So we define our null and alternative hypotheses to be:

$$
\begin{aligned}
H_0 &: \mu_1 = \mu_2 \\
H_A &: \mu_1 \neq \mu_2
\end{aligned}
$$

Since our null hypothesis involves equality we are using a two-sided test. So lets go ahead and compute the z-statistic noting that $\bar{\mu_1} = 639.6$ and $\bar{\mu_2} = 759.2$ are the known sample means, $\sigma_1 = 60$ and $\sigma_2 = 90$ are the known population standard deviations, and $n = 40$ and $m = 60$ are the sample sizes for married and unmarried people, respectively.

$$
\begin{aligned}
z &= \frac{\bar{\mu_1} - \bar{\mu_2} - \Delta\mu_0}{\sqrt{\sigma_1^2/n + \sigma_2^2/m}} \\
&= \frac{639.60 - 759.20 - 0}{\sqrt{60^2/40 + 90^2/60}} \\
&= -7.97.
\end{aligned}
$$

Now since we are using a two-sided test

$$p = 2p(Z < -|z|) = 2P(-|z|) = 2P(-|-7.97|)$$

Looking up the z-table in the formula sheet for $z = -|-7.97| = -7.97$ we see there are only p-values for z-values down to -3.49, which corresponds to $p = 0.0002$. So we can only conclude that $P(-|-7.97|) < 0.0002$. This means for our two-sided test we have

$$p = 2p(Z < -|z|) = 2P(-|z|) = 2P(-|-7.97|) < 0.0004$$

This is less than the typical significance value of $\alpha = 0.05$ so we conclude to reject the null hypothesis and instead except the alternative hypothesis which says married and unmarried people don't earn the same income.

R code hackers brain-busting challenge 1

Solve this problem using calculations in R and the relevant built in cdf function.

An economist was curious if women were more satisfied with their jobs than men. A random sample of 220 workers showed that 46 of 100 women were satisfied with their jobs, and 42 of 120 men were satisfied. Test the hypothesis that the proportion of women satisfied with their job is equal to the proportion of men satisfied with their job.

Answer

Assume θ_x is the proportion of women satisfied with jobs, θ_y is the proportion of men satisfied with their job. So the RVs X and Y will be binary since satisfaction is either yes I am satisfied or no I am not satisfied. The values the RVs can take on are either 1 or 0 which means the problem involves the Bernoulli RVs with population variance unknown.

We will test:

$$H_0 : \theta_x = \theta_y$$
$$H_A : \theta_x \neq \theta_y$$

This will be a two_sided test.

In [1]:

```
#Calculate based on the formula
satisfy_x <- 46
satisfy_y <- 42
count_x <- 100
count_y <- 120
theta_x <- satisfy_x/count_x
theta_y <- satisfy_y/count_y
# theta_p is the pooled estimate of the sample variance
theta_p <- (satisfy_x+satisfy_y)/(count_x+count_y)
z_value <- (theta_x-theta_y)/(sqrt(theta_p*(1-theta_p)*(1/count_x + 1/count_y)))
pval = 2 * pnorm(-abs(z_value))
result = ifelse(pval > 0.05,"we have weak/no evidence against the null", ifelse(pval<0.01,
                                                                                "we have strong evidence against the nul
                                                                                "we have moderate evidence against the n

cat("The p-value is:", pval, "\n")
cat("so,", result)
```

```
The p-value is: 0.09725443
so, we have weak/no evidence against the null
```

In the calculation above, we are using the formula from the lectures:

$$z_{(\hat{\theta}_x - \hat{\theta}_y)} = \frac{\hat{\theta}_x - \hat{\theta}_y}{\sqrt{\hat{\theta}_p(1-\hat{\theta}_p)(1/n + 1/m)}} \tag{1}$$

In the Formula Sheet (https://lms.monash.edu/mod/resource/view.php?id=7439150), you can also see the formula is listed as:

$$z_{(\hat{\theta}_x - \hat{\theta}_y)} = \frac{\hat{\theta}_x - \hat{\theta}_y - \Delta\theta_0}{\sqrt{\hat{\theta}_x(1-\hat{\theta}_x)/n + \hat{\theta}_y(1-\hat{\theta}_y)/m)}} \tag{2}$$

Formula (2) is provided for the general situation when the variances of the two sample groups are not known to be equal. Since we are using a two sided test and assuming $\theta_x = \theta_y$ is the null hypothesis which implys that the two groups' means are the same. As a result, the population variance of the binary RVs X and Y are also the same: $V[X] = \theta_x(1 - \theta_x) = \theta_y(1 - \theta_y) = V[Y]$. Therefore, we can use a pooled estimate of the sample variance $\hat{\theta}_p$ to replace $\hat{\theta}_x$ and $\hat{\theta}_y$ in the denominator of (2). Then in this case, we can use the formula (1) to get the z-value of this problem.

Hint to help you understand above:

According to the Bernoulli distribution, if X is a random variable with this distribution, then: $P(X = 1) = p = 1 - P(X = 0)$, so $P(X = 0) = 1 - p$. The EV of a Bernoulli random variable X is $E[X] = P(X = 1) \cdot 1 + P(X = 0) \cdot 0 = p \cdot 1 + (1 - p) \cdot 0 = p$. Also, we will know $E[X^2] = P(X = 1) \cdot 1^2 + P(X = 0) \cdot 0^2 = p \cdot 1^2 + (1 - p) \cdot 0^2 = p$. Therefore the variance $V[X]$ will be $E[x^2] - E[X]^2 = p - p^2 = p(1 - p)$.

You can also use other R functions to do more exact statistical tests (not covered in the lecture):

In [2]:

```r
# Using the prop.test() R to test difference between two Bernoulli samples
# x: a vector of counts satisfied
# n: a vector of counts observations
test_result <- prop.test(x = c(46, 42), n = c(100, 120), alternative = "two.sided")
print(test_result)
```

```
	2-sample test for equality of proportions with continuity correction

data:  c(46, 42) out of c(100, 120)
X-squared = 2.3108, df = 1, p-value = 0.1285
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.02887769  0.24887769
sample estimates:
prop 1 prop 2 
  0.46   0.35 
```

R code hackers brain-busting challenge 2: Comparing between two models' performance

For this question, we will use the logistic models taught to you in the previous week to compare between the performance of the two models. We will consider a full model and a fancy model with all feature interactions and transformations as shown below.

We will begin with reading the data (both training data and testing data)

```
In [2]: pima_train <- read.csv("pima_train.csv")
        pima_test <- read.csv("pima_test.csv")
```

Then we will define the function to calculate the model performance using `my.pred.stats()` function

In [3]:

```
# define the reporter function; copied from the R file on Moodle
my.pred.stats <- function(prob, target){
    classes = levels(target)
    # Convert probabilities to best guesses at classes
    pred = factor(prob > 1/2, c(F,T), classes)
    cat("---------------------------------------------------------------------------\n")
    cat("Performance statistics:\n")
    cat("\n")
    # cat("Confusion matrix:\n\n")
    T = table(pred,target)
    print(T)
    # cat("\n")
    cat("Classification accuracy =", mean(pred==target), "\n")
    cat("Sensitivity =", T[2,2]/(T[1,2]+T[2,2]), "\n")
    cat("Specificity =", T[1,1]/(T[1,1]+T[2,1]), "\n")
    # roc.obj = roc(response=as.numeric(target)-1, prob)
    # cat("Area-under-curve =", roc.obj$auc, "\n")
    # Prob is probability of success, so if the target is not a success, flip the probability
    # to get probability of failure
    prob[target==classes[1]] = 1 - prob[target==classes[1]]
    # Also make sure we never get exactly zero or one for probabilities due to numerical rounding
    prob = (prob+1e-10)/(1+2e-10)
    cat("Negative log-likelihood =", -sum(log(prob)), "\n")
    cat("Mean square error =", sum(prob*prob)/length(prob), "\n")
    cat("\n")
    # plot(roc.obj)
    cat("---------------------------------------------------------------------------\n")
    return(pred==target)
}
```

We will begin with building the full model

In [4]:
```
fullmod <- glm(DIABETES ~ . , data=pima_train, family=binomial)
summary(fullmod)
```

```

Call:
glm(formula = DIABETES ~ ., family = binomial, data = pima_train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.8045  -0.7175  -0.3920   0.7024   2.3013  

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -8.5271236  0.7988517 -10.674  < 2e-16 ***
PREG         0.1255938  0.0346542   3.624 0.000290 ***
PLAS         0.0353683  0.0043183   8.190  2.6e-16 ***
BP          -0.0170075  0.0071017  -2.395 0.016627 *  
SKIN         0.0136405  0.0153301   0.890 0.373582    
INS          0.0003532  0.0013082   0.270 0.787181    
BMI          0.0805829  0.0214811   3.751 0.000176 ***
PED          0.8410120  0.3293096   2.554 0.010653 *  
AGE          0.0189665  0.0104655   1.812 0.069943 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 868.88  on 667  degrees of freedom
Residual deviance: 618.08  on 659  degrees of freedom
AIC: 636.08

Number of Fisher Scoring iterations: 5
```

In [5]:

```
# Getting the model performance
full_accuracy <- my.pred.stats(predict(fullmod, pima_test, type="response"), pima_test$DIABETES)
```

```
---------------------------------------------------------------------------
Performance statistics:

    target
pred  N  Y
   N 61 15
   Y  8 16
Classification accuracy = 0.77
Sensitivity = 0.516129
Specificity = 0.884058
Negative log-likelihood = 49.58128
Mean square error = 0.5312035

---------------------------------------------------------------------------
```

Next, we will get the fancy model which include all input interactions and input transformations

In [6]:

```
fancymod = glm(DIABETES ~ . + .*. + log(PREG+1) + log(PLAS) + log(BP) + log(SKIN) + log(INS) + log(BMI) + log(PED) + log
summary(fancymod)
```

```
Call:
glm(formula = DIABETES ~ . + . * . + log(PREG + 1) + log(PLAS) + 
    log(BP) + log(SKIN) + log(INS) + log(BMI) + log(PED) + log(AGE) + 
    I(PREG^2) + I(PLAS^2) + I(BP^2) + I(SKIN^2) + I(INS^2) + 
    I(BMI^2) + I(PED^2) + I(AGE^2), family = binomial, data = pima_train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5067  -0.6278  -0.1744   0.6424   3.3494  

Coefficients:
                Estimate Std. Error z value Pr(>|z|)   
(Intercept)   -1.846e+02  9.118e+01  -2.025  0.04291 * 
PREG           2.513e-01  5.115e-01   0.491  0.62317   
PLAS          -1.951e-01  3.369e-01  -0.579  0.56258   
BP             2.126e-01  3.316e-01   0.641  0.52132   
SKIN          -7.893e-02  3.052e-01  -0.259  0.79591   
INS           -2.989e-02  2.487e-02  -1.202  0.22949   
BMI           -3.522e+00  1.273e+00  -2.768  0.00564 **
PED            4.334e+00  4.620e+00   0.938  0.34827   
AGE            1.235e+00  6.701e-01   1.842  0.06540 . 
log(PREG + 1)  3.127e-01  1.091e+00   0.287  0.77434   
log(PLAS)      1.612e+01  2.083e+01   0.774  0.43911   
log(BP)       -8.286e+00  9.459e+00  -0.876  0.38099   
log(SKIN)      3.423e+00  4.175e+00   0.820  0.41225   
log(INS)       2.778e+00  1.772e+00   1.568  0.11684   
log(BMI)       7.038e+01  2.376e+01   2.962  0.00306 **
log(PED)       9.198e-01  1.148e+00   0.801  0.42314   
log(AGE)      -1.570e+01  1.266e+01  -1.240  0.21482   
I(PREG^2)      2.230e-02  2.044e-02   1.091  0.27529   
I(PLAS^2)      5.900e-04  6.748e-04   0.874  0.38192   
I(BP^2)       -1.279e-03  1.337e-03  -0.957  0.33871   
I(SKIN^2)      1.492e-03  1.913e-03   0.780  0.43553   
I(INS^2)       1.279e-05  2.342e-05   0.546  0.58482   
I(BMI^2)       2.375e-02  9.187e-03   2.586  0.00972 **
I(PED^2)       9.852e-01  1.495e+00   0.659  0.50990   
```

```
I(AGE^2)      -7.883e-03  4.159e-03  -1.896  0.05802 .
PREG:PLAS     -2.233e-03  1.771e-03  -1.261  0.20734  
PREG:BP        8.031e-05  2.424e-03   0.033  0.97357  
PREG:SKIN      2.464e-03  6.498e-03   0.379  0.70455  
PREG:INS       1.921e-04  9.332e-04   0.206  0.83688  
PREG:BMI      -8.506e-04  8.623e-03  -0.099  0.92142  
PREG:PED       1.734e-01  1.310e-01   1.324  0.18550  
PREG:AGE      -9.977e-03  4.562e-03  -2.187  0.02873 *
PLAS:BP        1.524e-04  4.202e-04   0.363  0.71688  
PLAS:SKIN      8.710e-04  7.178e-04   1.213  0.22494  
PLAS:INS       5.772e-05  9.783e-05   0.590  0.55520  
PLAS:BMI      -1.038e-03  1.106e-03  -0.939  0.34793  
PLAS:PED      -2.787e-02  1.569e-02  -1.777  0.07564 .
PLAS:AGE      -1.140e-03  5.834e-04  -1.953  0.05079 .
BP:SKIN       -1.032e-03  1.509e-03  -0.684  0.49409  
BP:INS        -2.311e-04  1.589e-04  -1.454  0.14594  
BP:BMI         1.096e-03  1.673e-03   0.655  0.51251  
BP:PED         3.077e-02  2.895e-02   1.063  0.28775  
BP:AGE         2.015e-03  1.024e-03   1.969  0.04901 *
SKIN:INS       6.199e-05  2.451e-04   0.253  0.80031  
SKIN:BMI      -6.106e-04  3.944e-03  -0.155  0.87697  
SKIN:PED       3.588e-03  5.983e-02   0.060  0.95218  
SKIN:AGE      -4.421e-03  2.091e-03  -2.115  0.03446 *
INS:BMI        3.320e-05  4.039e-04   0.082  0.93451  
INS:PED       -3.413e-03  4.211e-03  -0.811  0.41759  
INS:AGE        5.219e-04  2.965e-04   1.760  0.07834 .
BMI:PED       -7.062e-02  7.816e-02  -0.904  0.36622  
BMI:AGE       -4.523e-04  2.549e-03  -0.177  0.85919  
PED:AGE       -8.590e-02  3.863e-02  -2.224  0.02616 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 868.88  on 667  degrees of freedom
Residual deviance: 533.75  on 615  degrees of freedom
AIC: 639.75

Number of Fisher Scoring iterations: 7
```

In [7]:

```
# Getting the model performance
fancy_accuracy <- my.pred.stats(predict(fancymod, pima_test, type="response"), pima_test$DIABETES)
```

```
---------------------------------------------------------------------------
Performance statistics:

    target
pred  N  Y
   N 59 12
   Y 10 19
Classification accuracy = 0.78
Sensitivity = 0.6129032
Specificity = 0.8550725
Negative log-likelihood = 46.88369
Mean square error = 0.5877172

---------------------------------------------------------------------------
```

This model includes a lot more features compared to the first one.

YOUR TASK is to statistically compare the performance between the two models using hypothesis testing. In order to do this, the easiest way would be to compare between the two prediction sets coming from the two models (prediction sets are acquired using the test set).

Answer

Our hypothesis is to compare the prediction performance between the two model, in this case, the performance will be denoted as θ

$$H_0 : \theta_{\text{full}} = \theta_{\text{fancy}}$$
$$H_A : \theta_{\text{full}} \neq \theta_{\text{fancy}}$$

We will use the same formula listed in the question above:

$$z_{(\hat{\theta}_x - \hat{\theta}_y)} = \frac{\hat{\theta}_x - \hat{\theta}_y}{\sqrt{\hat{\theta}_p(1 - \hat{\theta}_p)(1/n + 1/m)}} \tag{1}$$

While in the Formula Sheet (https://lms.monash.edu/mod/resource/view.php?id=7439150), the formula is listed as:

$$z_{(\hat{\theta}_x - \hat{\theta}_y)} = \frac{\hat{\theta}_x - \hat{\theta}_y - \Delta\theta_0}{\sqrt{\hat{\theta}_x(1-\hat{\theta}_x)/n + \hat{\theta}_y(1-\hat{\theta}_y)/m)}} \tag{2}$$

Disclamer: to perform a proper hypothesis testing here, we also need to take into account the number of features used in each model to perform the analysis; however, for the sake of simplification, we will ignore that and just compare two prediction vectors.

We then will calculate $\hat{\theta}_p = \frac{m_x+m_y}{n_x+n_y} = \frac{77+78}{100+100} = 0.775$ as seen from below, the $z_{\text{value}} \approx -0.17$ this z_{value} is not significant enough to reject the hypothesis, meaning that the two models are no different to each other and the fancy model doesn't capture any meaningful information to improve the full model.

In [8]:

```
m_x = sum(full_accuracy)
m_y = sum(fancy_accuracy)
n_x = length(full_accuracy)
n_y = length(fancy_accuracy)

cat(m_x, m_y, n_x, n_y)
cat("\n") # Breaking the line

theta_p = (m_x + m_y)/(n_x + n_y)
theta_x = mean(full_accuracy)
theta_y = mean(fancy_accuracy)

z_value = (theta_x - theta_y)/sqrt(theta_p*(1-theta_p)*(1/n_x + 1/n_y))

pval = 2 * pnorm(-abs(z_value))
result = ifelse(pval > 0.05,"we have weak/no evidence against the null", ifelse(pval<0.01,
                                                                                "we have strong evidence against the nul
                                                                                "we have moderate evidence against the n

cat("The z-value is:", z_value,"The p-value is:", pval, "\n")
cat("so,", result)
```

```
77 78 100 100
The z-value is: -0.1693335 The p-value is: 0.8655343
so, we have weak/no evidence against the null
```