ST 541 Project Report

Corey Williams

December 2023

1 Introduction

One of the most common tasks for a student in a statistics course is to assess the assumptions of a statistical test. These assumptions can vary from test to test but there are typically diagnostic plots that go along with each assumption like a qq-plot to assess normality or side by side boxplots to assess equal variance. Students are given these plots and told to assess the assumptions of the test but are often not given enough guidance on how to judge these plots. I remember in my first statistics course that I saw a qq-plot the explanation given was that "If the points closely follow the line then the data are normally distributed", while true this is somewhat unhelpful as no randomly generated points will perfectly fall along a qq-plot line. This project aims to create resources that can be used to help students better understand the assumptions of t-tests and ANOVA. This will include examples of diagnostic plots as well as assessments of how far assumptions can be stretched.

2 Methods

10,000 runs of each simulation were used to find the results described below. Sample sizes of 10, 30, and 100 were chosen for this project. These sample sizes were selected to be reflective of 3 common scenarios. 10 for a low sample size, 30 to test the rule of thumb given to students, and 100 for a large sample size. In the case where multiple groups are used these sample size refer to the number of observations per group. A significance level of $\alpha = 0.05$ was used for all tests performed so a type I error rate of 0.05 should be expected when all assumptions of a test are met.

2.1 Assumptions

Both the t-test and ANOVA share the same main assumptions, for brevity the assumptions will only be listed once. Note that in this wording groups will refer to the two samples in two-sample t-tests and the groups in ANOVA. The normality and equal variance assumptions are the assumptions explored in this project, independence was left for future work.

1. Independence) Observations are independent within and between groups.
2. Normality) Observations within a group come from a normal distribution.
3. Equal Variance) Variances of groups are equal. This assumption only applies to t-tests when Welch's t-test is not used.

To violate assumptions of normality the χ^2 distribution family was used with varying degrees of freedom. This family of distributions was used because as the degrees of freedom increase the distribution approaches a normal distribution. Degrees of freedom 1, 2, 3, 5, and 10 were used to provide a range of skewness with 1 being the most skewed and 10 being the least. By the time that the degrees of freedom reach 10 the χ^2 distribution reasonably satisfies the approximate normality assumption of t-tests and ANOVA.

To violate the equal variance assumption in t-tests one group had a constant standard deviation of 1 while the other group had its standard deviation varied between 1, 1.25, 2, and 5. For ANOVA three sets of non-equal variance were used along with a set of equal variances to serve as a control.

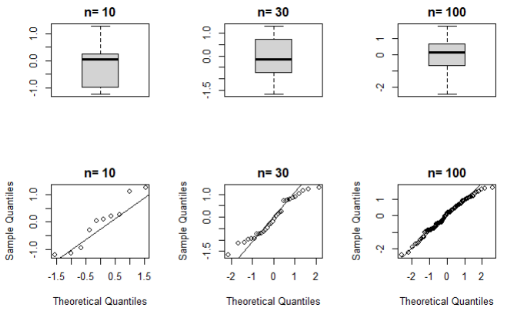

Figure 1: Assumption plots for Normal data

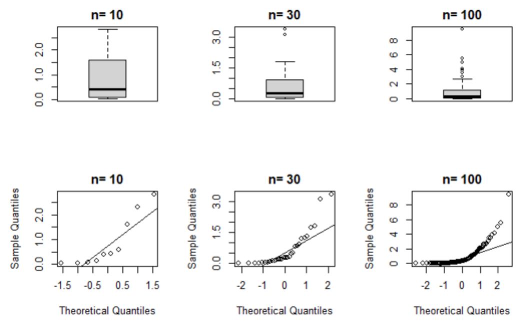

Figure 2: Assumption plots for $\chi^2_{(1)}$ data

3 Results

3.1 t-tests

In the one sample t-test the assumption of normality is hard to assess when the sample size is only 10. Both the boxplots and qqplots of normal and χ^2 distributions can look visually similar with high probability. When the sample size is increased to 30 assessing normality visually becomes much more consistent. In table 1 we can see the expected results that as normality is approached and sample size increases the type I error rate will approach the desired 0.05

n	df=1	df=2	df=3	df=5	df=10
10	0.14	0.10	0.09	0.07	0.06
30	0.09	0.07	0.06	0.06	0.05
100	0.07	0.06	0.05	0.05	0.05

Table 1: Type I error rates for sample size n (rows) and df (columns)

When looking at the normality assumption in two sample t-tests we can see that interestingly it did not matter if one or two samples violated the normality assumption with the distributions used. I would not expect for this result to hold for every distribution. In the case of one normal distribution and one χ^2 distribution a normal distribution with equal variance and expected value to the χ^2 distribution was used.

n	df=1	df=2	df=3	df=5	df=10
10	0.03	0.04	0.04	0.04	0.05
30	0.04	0.05	0.05	0.05	0.05
100	0.05	0.05	0.05	0.05	0.05

Table 2: Type I error rates when one sample follows a χ^2 distribution

n	df=1	df=2	df=3	df=5	df=10
10	0.03	0.04	0.04	0.04	0.05
30	0.04	0.05	0.05	0.05	0.05
100	0.05	0.05	0.05	0.05	0.05

Table 3: Type I error rates when both samples follow a χ^2 distribution

We can also examine what happens when the assumption of equal variance is violated. The results seen in tables 4 and 5 are to be expected, one interesting result to consider is that the type I error rate is hardly influenced by unequal variances. Even when one variance was $4\times$ the other the type I error rate remained 0.05.

n	$\sigma_2 = 1$	$\sigma_2 = 1.25$	$\sigma_2 = 2$	$\sigma_2 = 5$
10	0.05	0.05	0.05	0.06
30	0.05	0.05	0.05	0.06
100	0.05	0.05	0.05	0.05

Table 4: Type I error rates when samples have unequal variance $\sigma_1 = 1$

n	$\sigma_2 = 1$	$\sigma_2 = 1.25$	$\sigma_2 = 2$	$\sigma_2 = 5$
10	0.56	0.46	0.28	0.10
30	0.96	0.92	0.67	0.19
100	1.00	1.00	0.99	0.50

Table 5: Power when samples have unequal variance $\sigma_1 = 1$, $\mu_1 = 0$, and $\mu_2 = 1$

3.2 ANOVA

For ANOVA we can also violate the assumption of equal variance. This assumption can be violated in many possible ways since we could have many groups. The simulations performed examine what happens in the cases where; 1) one group has unequal variance, 2) there are 2 pairs of groups with equal variances, and 3) all groups have unequal variances.

n	sds=1, 1, 1, 1	sds=1, 1, 1, 2	sds=1, 2, 1, 2	sds=1, 2, 3, 4
10	0.05	0.06	0.07	0.07
30	0.05	0.06	0.06	0.07
100	0.05	0.06	0.06	0.07

Table 6: Type I error rates when samples have unequal variances

n	sds=1, 1, 1, 1	sds=1, 1, 1, 2	sds=1, 2, 1, 2	sds=1, 2, 3, 4
10	0.71	0.45	0.31	0.15
30	1.00	0.95	0.84	0.35
100	1.00	1.00	1.00	0.87

Table 7: Power when samples have unequal variances and means (1,1,2,2)

From these simulations it is clear to see that as the number of groups with unequal variance increases the lower power and the higher type I error rate we will observe. This is the expected result but it is interesting to see the significant drop in power of only one group having an unequal variance in the 10 units per group case. The simulations for ANOVA took substantially longer to run than the corresponding simulations for t-tests. I believe that this is due to a combination of the extra data generation and slightly more complex calculations required for ANOVA. It would be interesting to look at what improvements could be made in efficiency for simulating ANOVA. One idea that I had was rather than generating normally distributed data each iteration it may be sufficient to generate a large dataset of normally distributed data to sample from. These two would not be equivalent but it may be a reasonable approximation and may help save time during data generation (I am not sure what the relative efficiency of sampling vs generating normally distributed data is).

4 Conclusion

Based on the results from this project there are three pieces of advice I would give to students trying to assess the assumptions of a statistical test.

1. Be careful trying to assess the normality assumption at low sample sizes. Based only on visual inspection it is nearly impossible to identify normally distributed data at a sample size of 10. This means that additional thought should be put in to this assumption beyond just using plots, we should think about the mechanism that is generating the data. Would we expect the mechanism to generate symmetric or skewed data?

2. Assumptions will almost never be perfectly met. The only cases where we can know whether assumptions are fully met are in simulation studies like this one so for the most part the best we can do is stating whether we believe assumptions are approximately met. This is fine since as we have seen there are many scenarios where meeting assumptions approximately results in completely reasonable test results.

3. If you believe you are in violation of an assumption try to simulate a similar scenario. This piece of advice may be beyond the capabilities of many students but I believe it is still important advice. When we think we are in violation of an assumption a great way to understand that potential consequences of violating that assumption is to simulate data that is similar to the observed data. This allows us to understand what kind of power and type I error we should expect which would in turn allow us to determine if we think the test is useful.

There are two areas I would still like to explore further that were unfortunately not able to be covered in this project due to time constraints, violations of independence and violations of normality for ANOVA.

5 Appendix

5.1 Libraries and plots

```
library(tidyverse)

#### creating assumption plots
### Normal distribution
set.seed(pi)
n1<-10
n2<-30
n3<-100
x1<-rnorm(n1)
x2<-rnorm(n2)
x3<-rnorm(n3)
par(mfrow=c(2,3))
# boxplots
boxplot(x1,main=paste("n=",n1))
boxplot(x2,main=paste("n=",n2))
boxplot(x3,main=paste("n=",n3))
# qq plots
qqnorm(x1,main=paste("n=",n1))
qqline(x1)
qqnorm(x2,main=paste("n=",n2))
qqline(x2)
qqnorm(x3,main=paste("n=",n3))
qqline(x3)

### chisq distribution
x1<-rchisq(n1,1)
x2<-rchisq(n2,1)
x3<-rchisq(n3,1)
par(mfrow=c(2,3))
# boxplots
boxplot(x1,main=paste("n=",n1))
boxplot(x2,main=paste("n=",n2))
boxplot(x3,main=paste("n=",n3))
# qq plots
qqnorm(x1,main=paste("n=",n1))
qqline(x1)
qqnorm(x2,main=paste("n=",n2))
qqline(x2)
qqnorm(x3,main=paste("n=",n3))
qqline(x3)
```

5.2 One Sample t-tests

5.2.1 Non-Normality

```
# create dataframe of df and n combinations to use
type_I_errors<-data.frame(degrees_of_freedom=rep(c(1,2,3,5,10),times=3),
                          n=rep(c(10,30,100),each=5),
                          error_rate=rep(NA,15))
sim_t.tests<-function(nsim=10000,n=30,chisq=T,df=1){
  # returns the type I error rate for n t tests
  if(chisq==T){
```

```
    mean(map_dbl(1:nsim,~t.test(rchisq(n,df),mu=df)$p.value<0.05))
  }
  else{
    mean(map_dbl(1:nsim,~t.test(rnorm(n))$p.value<0.05))
  }
}
# Find type I errors
results<-with(type_I_errors,
     map2(as.list(degrees_of_freedom),as.list(n),~sim_t.tests(n=.y,df=.x)))
type_I_errors$error_rate<-unlist(results)
type_I_errors
# go from long to wide format to make a table
tab<-pivot_wider(type_I_errors, names_from = "degrees_of_freedom",
           values_from = "error_rate")
write_rds(tab,file="1-sample-t-test-error-rates")
```

5.3 Two Sample t-tests

5.3.1 Non-Normality

```
sim.two.samp.t.tests<-function(nsim=10000,n=30,chisq=T,both=T,df=1){
  # returns the type I error rate for n t tests
  if(chisq==T){ # check if we are violating normality
    if(both==T){ # check if one or both samples
      mean(map_dbl(1:nsim,
                   ~t.test(rchisq(n,df),rchisq(n,df))$p.value<0.05))
    }
    else{ # E(chisq)=df var(chisq)=2*df, use appropriate values for normal
      mean(map_dbl(1:nsim,
                   ~t.test(rchisq(n,df),rnorm(n,df,sqrt(2*df)))$p.value<0.05))
    }
  }
  else{
    mean(map_dbl(1:nsim,~t.test(rnorm(n),rnorm(n))$p.value<0.05))
  }
}
df<-c(1,2,3,5,10)
n<-c(10,30,100)
column_names<-c("degrees_of_freedom","n")
# create data frames of df and n combinations to iterate over
one_chisq<-data.frame(expand.grid(df,n))
colnames(one_chisq)<-column_names
both_chisq<-as.data.frame(cbind(expand.grid(df,n)))
colnames(both_chisq)<-column_names

# simulate the one chisq one normal distribution
# map each combination of df and n into function to find type I error rate
results_one_chisq<-with(one_chisq,
     map2(as.list(degrees_of_freedom),as.list(n),
          ~sim.two.samp.t.tests(n=.y,df=.x)))
one_chisq$error_rate<-unlist(results_one_chisq)
tab_1_chisq<-pivot_wider(one_chisq,
                         names_from = "degrees_of_freedom",
                         values_from = "error_rate")
write_rds(tab_1_chisq,file="2-sample-t-test-error-rates-1chisq")

# simulate the both chisq distributions
```

```
results_both_chisq<-with(both_chisq,
     map2(as.list(degrees_of_freedom),as.list(n),
          ~sim.two.samp.t.tests(n=.y,df=.x)))
both_chisq$error_rate<-unlist(results_both_chisq)
tab_2_chisq<-pivot_wider(both_chisq,
                         names_from = "degrees_of_freedom",
                         values_from = "error_rate")
write_rds(tab_2_chisq,file="2-sample-t-test-error-rates-2chisq")
```

5.3.2 Non-Equal Variance

```
######## Type I error
sim.unequal.var<-function(n=30,nsim=10000,sd1=1,sd2=2){
  # returns the type I error rate of nsim simulations of equal var t test
  mean(map_lgl(1:nsim,
               ~t.test(rnorm(n=n,0,sd1),
                       rnorm(n=n,0,sd2),
                       var.equal=T)$p.value<0.05))
}

sd2<-c(1,1.25,2,5)
n<-c(10,30,100)
# create dataframe to store resulst and iterate over
results_uneq_var<-expand.grid(n,sd2)
colnames(results_uneq_var)<-c("n","sd2")
# perform simulation
error_rates_uneq<-map2(results_uneq_var$n, results_uneq_var$sd2,
                        ~sim.unequal.var(n=.x,sd2=.y))
# save results into dataframe
results_uneq_var$error_rate<-unlist(error_rates_uneq)
tab_uneq<-pivot_wider(results_uneq_var,
                         names_from = "sd2",
                         values_from = "error_rate")
write_rds(tab_uneq,file="2-sample-t-test-error-rates-uneq")

######## Power
sim.unequal.var.power<-function(n=30,nsim=10000,sd1=1,sd2=2){
  # returns the type I error rate of nsim simulations of equal var t test
  mean(map_lgl(1:nsim,
               ~t.test(rnorm(n=n,0,sd1),
                       rnorm(n=n,1,sd2),
                       var.equal=T)$p.value<0.05))
}
# create dataframe to store resulst and iterate over
power_results<-expand.grid(n,sd2)
colnames(power_results)<-c("n","sd2")
# perform simulation
power_uneq<-map2(power_results$n, power_results$sd2,
                        ~sim.unequal.var.power(n=.x,sd2=.y))
# save results into dataframe
power_results$error_rate<-unlist(power_uneq)
tab_uneq_power<-pivot_wider(power_results,
                         names_from = "sd2",
                         values_from = "error_rate")
write_rds(tab_uneq_power,file="2-sample-t-test-power-uneq")
```

5.4 ANOVA

5.4.1 Non-Equal Variance

```
######## Type I error
n<-c(10,30,100)
sds<-list(c(1,1,1,1), c(1,2,3,4), c(1,2,1,2), c(1,1,1,2))
mus<-list(c(1,1,1,1))
groups<-c("a","b","c","d")

sim_anova_errors<-function(nsim=10000, n=30, mus=c(1,1,1,1),sds=c(1,1,1,1),
                           groups=c("a","b","c","d")){
  # finds the type I error rate of an anova with the specified means & sds
  map_lgl(1:nsim,
          ~anova(aov(data=data.frame(groups=rep(groups, each=n),
                                     values=c(rnorm(n,mus[1],sds[1]),
                                              rnorm(n,mus[2],sds[2]),
                                              rnorm(n,mus[3],sds[3]),
                                              rnorm(n,mus[4],sds[4]))),
                values~groups))$`Pr(>F)`[1]<0.05)%>%
    mean()
}

# create dataframe to store resulst and iterate over
anova_errors<-expand.grid(n,sds)
colnames(anova_errors)<-c("n","sds")
# perform simulation
anova_results<-map2_dbl(anova_errors$n, anova_errors$sds,
                        ~sim_anova_errors(n=.x,sds=.y))
anova_errors$Type_I_error_rate<-anova_results
# go from long to wide format
anova_uneq_error_table<-pivot_wider(anova_errors,
                         names_from = "sds",
                         values_from = "Type_I_error_rate")
# save object so I don't have to rerun
write_rds(anova_uneq_error_table,file="anova_uneq_error_table")

######## Power
sim_anova_power<-function(nsim=10000, n=30, mus=c(1,1,2,2),sds=c(1,1,1,1),
                          groups=c("a","b","c","d")){
  # finds the type I error rate of an anova with the specified means & sds
  map_lgl(1:nsim,
          ~anova(aov(data=data.frame(groups=rep(groups, each=n),
                                     values=c(rnorm(n,mus[1],sds[1]),
                                              rnorm(n,mus[2],sds[2]),
                                              rnorm(n,mus[3],sds[3]),
                                              rnorm(n,mus[4],sds[4]))),
                values~groups))$`Pr(>F)`[1]<0.05)%>%
    mean()
}

# create dataframe to store resulst and iterate over
anova_power<-expand.grid(n,sds)
colnames(anova_power)<-c("n","sds")
# perform simulation
anova_results<-map2_dbl(anova_power$n, anova_power$sds,
                        ~sim_anova_power(n=.x,sds=.y))
anova_power$Power<-anova_results
# go from long to wide format
```

```
anova_uneq_power_table<-pivot_wider(anova_power,
                      names_from = "sds",
                      values_from = "Power")
write_rds(anova_uneq_power_table,file="anova_uneq_power_table")
```