✪ Psychological Statistics ✪

Week 04: *Comparing means of 2 groups (Parametric)*

- Edited by Prof. **Changwei Wu**
- Graduate Institute of Mind, Brain and Consciousness (***GIMBC***), Taipei Medical University

In [ ]:

```
### [ Setup the working directory ]

setwd("/Users/wesley/[Course]/Python/R_Script")
getwd()
```

In [18]:

```
### [ Loading the required libraries ]

library("tidyverse")
library("rstatix")
library("effsize")
library("ggplot2")
library("ggpubr")
```

---

(1) Pipe functions {dplyr}

In [65]:

```
### 1-1.[ Use %>% for transfer output to the next level ]

# → Loading the dataset
 data(diamonds)

# → using pipeline
dim(head(diamonds,4))

diamonds %>% head(4) %>% dim
```

4 · 10

4 · 10

In [63]:

```
### 1-2.[ To select specific column (vector) from a 2-D data frame by "select"]
# head(diamonds,5)

# → Only select 2 columns: Carat & Price
diamonds %>% select(carat, price) %>% head(3)
# diamonds[, c('carat','price')] # → Traditional way


# → NOT to select 2 columns: Carat & Price
diamonds %>% select(-carat, -price) %>% head(3)
```

A tibble: 3 × 2

carat	price
<dbl>	<int>
0.23	326
0.21	326
0.23	327

A tibble: 3 × 8

cut	color	clarity	depth	table	x	y	z
<ord>	<ord>	<ord>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Ideal	E	SI2	61.5	55	3.95	3.98	2.43
Premium	E	SI1	59.8	61	3.89	3.84	2.31
Good	E	VS1	56.9	65	4.05	4.07	2.31

In [62]:

```
### 1-3.[ To select the row values in a certain column (vector) by "filter" ]

#--- Traditional way to choose ideal cut ---#
#diamonds[diamonds$cut == 'Ideal', ]

# → pipe function
diamonds %>% filter(cut == 'Ideal') %>% head(3)

diamonds %>% filter(cut %in% c('Ideal','Good')) %>% head(3)

diamonds %>% filter(carat > 2 & price < 6000)

# diamonds$carat <- ifelse(diamonds$carat > 1, 1, diamonds$carat)
```

A tibble: 3 × 10

carat	cut	color	clarity	depth	table	price	x	y	z
<dbl>	<ord>	<ord>	<ord>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.23	Ideal	J	VS1	62.8	56	340	3.93	3.90	2.46
0.31	Ideal	J	SI2	62.2	54	344	4.35	4.37	2.71

A tibble: 3 × 10

carat	cut	color	clarity	depth	table	price	x	y	z
<dbl>	<ord>	<ord>	<ord>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75

A tibble: 7 × 10

carat	cut	color	clarity	depth	table	price	x	y	z
<dbl>	<ord>	<ord>	<ord>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
2.06	Premium	J	I1	61.2	58	5203	8.10	8.07	4.95
2.14	Fair	J	I1	69.4	57	5405	7.74	7.70	5.36
2.15	Fair	J	I1	65.5	57	5430	8.01	7.95	5.23
2.22	Fair	J	I1	66.7	56	5607	8.04	8.02	5.36
2.01	Fair	I	I1	67.4	58	5696	7.71	7.64	5.17
2.01	Fair	I	I1	55.9	64	5696	8.48	8.39	4.71
2.27	Fair	J	I1	67.6	55	5733	8.05	8.00	5.43

In [6]:

```
### 1-4.[ To create a new vector by "mutate" ]
# → Additional function: arrange [to perform sorting for different row entries]

diamonds %>% select(carat, price) %>% mutate(price/carat) %>% head(3)

diamonds %>% select(carat, price) %>%
            mutate(Ratio=price/carat, Double=Ratio*2) %>% arrange(Ratio) %>% head(3)
```

A tibble: 3 × 3

carat	price	price/carat
<dbl>	<int>	<dbl>
0.23	326	1417.391
0.21	326	1552.381
0.23	327	1421.739

A tibble: 3 × 4

carat	price	Ratio	Double
<dbl>	<int>	<dbl>	<dbl>
0.43	452	1051.163	2102.326
0.32	345	1078.125	2156.250
0.31	335	1080.645	2161.290

In [17]:

```
### 1-5.[ To conduct analysis by categorical levels in a vector by "group_by"]
# → Additional function: summarize [to calculate the descriptive stat. by function r

diamonds %>% summarize(mean(price))

diamonds %>% group_by(cut) %>%
    summarize(AvgPrice=mean(price), SumCarat=sum(carat))

# aggregate(price~cut, diamonds, mean) # Traditional way
```

A tibble: 1 × 1

mean(price)
<dbl>
3932.8

A tibble: 5 × 3

cut	AvgPrice	SumCarat
<ord>	<dbl>	<dbl>
Fair	4358.758	1684.28
Good	3928.864	4166.10
Very Good	3981.760	9742.70
Premium	4584.258	12300.95
Ideal	3457.542	15146.84

In [61]:

```
### 1-6.[ To change the wide form to long form through pipe function]
# → Function: gather {tidyr}

data("mice2", package = "datarium")
head(mice2, 3)

#----- Data restructure (Wide → Long) -----#
mice2.long <- mice2 %>%
    gather(key = "group", value = "weight", before, after)

head(mice2.long, 3)
```

A data.frame: 3 × 3

	id	before	after
	<int>	<dbl>	<dbl>
1	1	187.2	429.5
2	2	194.2	404.4
3	3	231.7	405.6

A data.frame: 3 × 3

	id	group	weight
	<int>	<chr>	<dbl>
1	1	before	187.2
2	2	before	194.2
3	3	before	231.7

(2) Student's *t* Test

Example A: Spider Anxiety (as of one single group)

[Hypothesis] The anxiety level is different from 0. *(2-tailed)*

- Null hypothesis H_0 : Anxiety(spider) = 0
- Alternative hyp. H_1 : Anxiety(spider) ≠ 0

In [149]:

```
### [ Step.1 ] Data loading and Descriptive stat.

#----- Data preparation -----#

Spider <- read.delim("SpiderLong.dat")

#----- Desctiptive stat. -----#
#Spider %>% get_summary_stats(Anxiety)

Spider %>% get_summary_stats(Anxiety, type = "mean_sd")
```

A tibble: 1 × 4

variable	n	mean	sd
<chr>	<dbl>	<dbl>	<dbl>
Anxiety	24	43.5	10.595

In [150]:

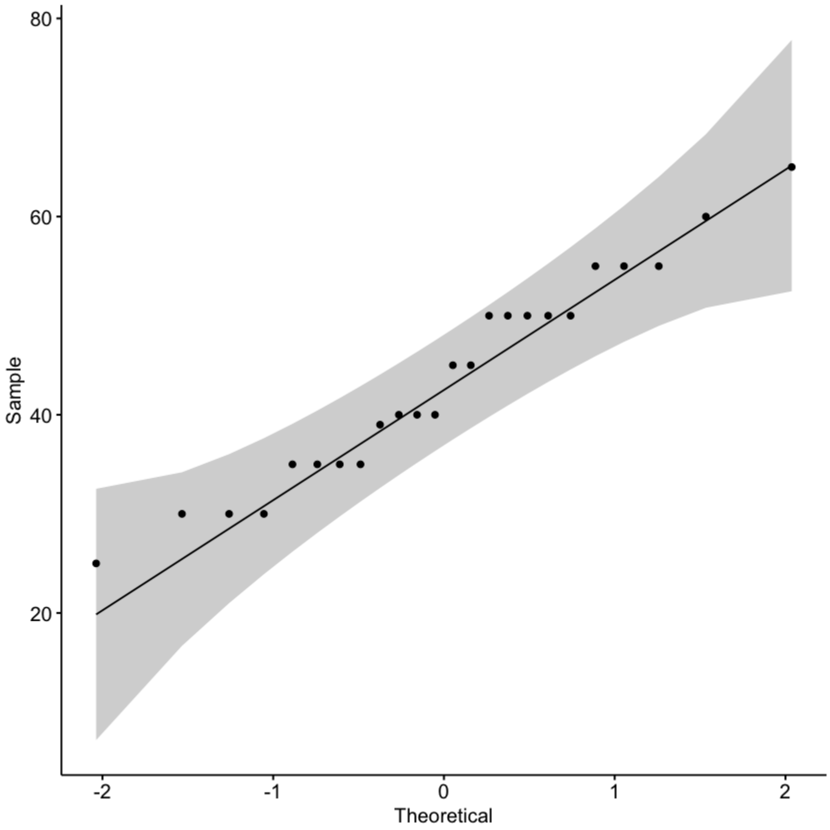
```
### [ Step.2 ]  Assumption check

#----- (a) Outliers -----#
Spider %>% identify_outliers(Anxiety)

#----- (b) Normality -----#
Spider %>% shapiro_test(Anxiety)

ggqqplot(Spider, x = "Anxiety")
```

A data.frame: 0 × 4

Group	Anxiety	is.outlier	is.extreme
<fct>	<int>	<lgl>	<lgl>

A tibble: 1 × 3

variable	statistic	p
<chr>	<dbl>	<dbl>
Anxiety	0.9628216	0.4976814



In [7]:

```
### [ Step.3 ] one-sample t-test

(OneSamp.test <- Spider %>% t_test(Anxiety ~ 1, mu = 0))
```

A rstatix_test: 1 × 7

	.y.	group1	group2	n	statistic	df	p
	<chr>	<chr>	<chr>	<int>	<dbl>	<dbl>	<dbl>
1	Anxiety	1	null model	24	20.11318	23	4.28e-16

In [132]:

```
### [ Step.4 ] Effect size

Spider %>% cohens_d(Anxiety ~ 1, mu = 0)
```

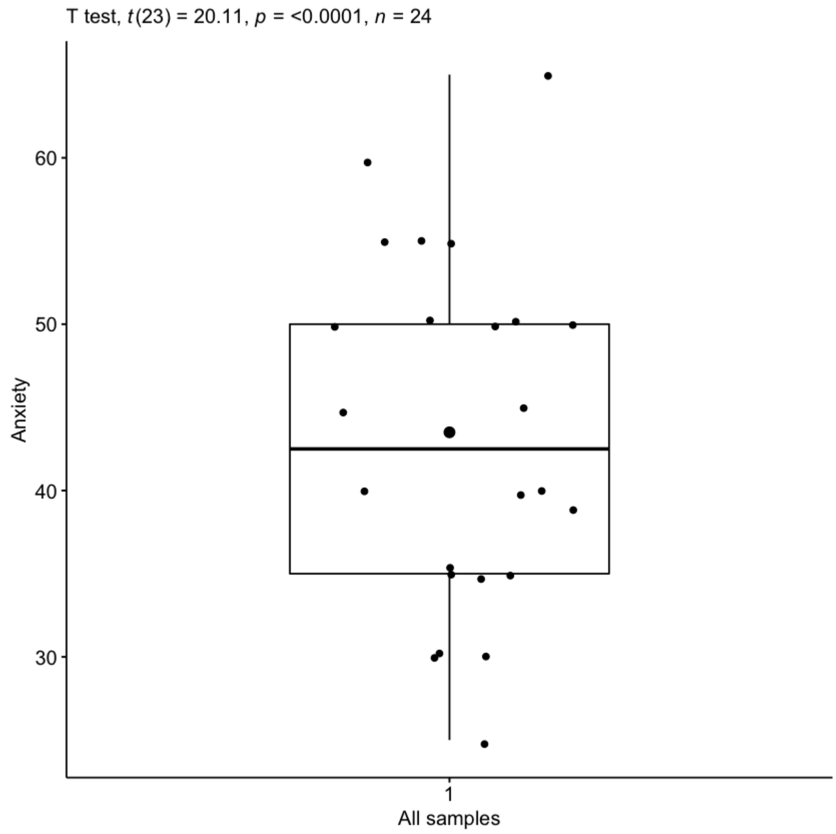
A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
1	Anxiety	1	null model	4.105585	24	large

In [135]:

```
### [ Step.5 ] Visualization & Report

ggboxplot(Spider, y="Anxiety", width = .5, add = c("mean","jitter"),
          ylab = "Anxiety", xlab = "All samples") +
    labs(subtitle = get_test_label(OneSamp.test, detailed = TRUE))
```

```
Warning message:
"`fun.y` is deprecated. Use `fun` instead."
Warning message:
"`fun.ymin` is deprecated. Use `fun.min` instead."
Warning message:
"`fun.ymax` is deprecated. Use `fun.max` instead."
```



~ *Report* ~

- **A one-sample t-test exhibited that the measured anxiety level (43.5 ± 10.6) was statistically significantly higher than 0 (t_{23} = 20.1, p < .001, d = 4.1).**
- [Optional]: The anxiety level was normally distributed, as assessed by Shapiro-Wilk's test (p > 0.05) and there were no extreme outliers in the data.

Example B: Spider Anxiety (as of 2 independent groups)

[Hypothesis] Anxiety of viewing real spider is different from anxiety of viewing spider picture. *(2-tailed)*

- Null hypothesis H_0 **:** Anxiety(real spider) = Anxiety(spider picture)
- Alternative hyp. H_1 **:** Anxiety(real spider) ≠ Anxiety(spider picture)

In [9]:

```
### [ Step.1 ] Data loading and Descriptive stat.

#----- Data preparation -----#

Spider <- read.delim("SpiderLong.dat")
levels(Spider$Group)

#----- Reorder the Levels -----#
Spider$Group <- factor(Spider$Group, levels = c("Real Spider", "Picture"))
levels(Spider$Group)

#----- Rename the Levels -----#
(levels(Spider$Group) <-  c("Real", "Picture"))


#----- Desctiptive stat. -----#
#Spider %>% get_summary_stats(Anxiety)

Spider %>% group_by(Group) %>%
    get_summary_stats(Anxiety, type = "mean_sd")
```

'Picture' · 'Real Spider'

'Real Spider' · 'Picture'

'Real' · 'Picture'

A tibble: 2 × 5

Group	variable	n	mean	sd
<fct>	<chr>	<dbl>	<dbl>	<dbl>
Real	Anxiety	12	47	11.029
Picture	Anxiety	12	40	9.293

In [10]:

```
### [ Step.2 ]  Assumption check

#----- (a) Outliers -----#
Spider %>% group_by(Group) %>%
    identify_outliers(Anxiety)

#----- (b) Normality -----#
Spider %>% group_by(Group) %>%
    shapiro_test(Anxiety)

#ggqqplot(Spider, x = "Anxiety", facet.by = "Group")

#----- (c) Homogeneity -----#
Spider %>% levene_test(Anxiety ~ Group)
```

A data.frame: 0 × 4

Group	Anxiety	is.outlier	is.extreme
<fct>	<int>	<lgl>	<lgl>

A tibble: 2 × 4

Group	variable	statistic	p
<fct>	<chr>	<dbl>	<dbl>
Real	Anxiety	0.9488729	0.6205694
Picture	Anxiety	0.9650165	0.8522870

A tibble: 1 × 4

df1	df2	statistic	p
<int>	<int>	<dbl>	<dbl>
1	22	0.2990654	0.5899734

In [11]:

```
### [ Step.3 ] 2-sample t-test

(TwoSamp.test <- Spider %>%
     t_test(Anxiety ~ Group, paired = FALSE) %>% add_significance())
```

A rstatix_test: 1 × 9

.y.	group1	group2	n1	n2	statistic	df	p	p.signif
<chr>	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>	<chr>
Anxiety	Real	Picture	12	12	1.681346	21.38502	0.107	ns

In [12]:

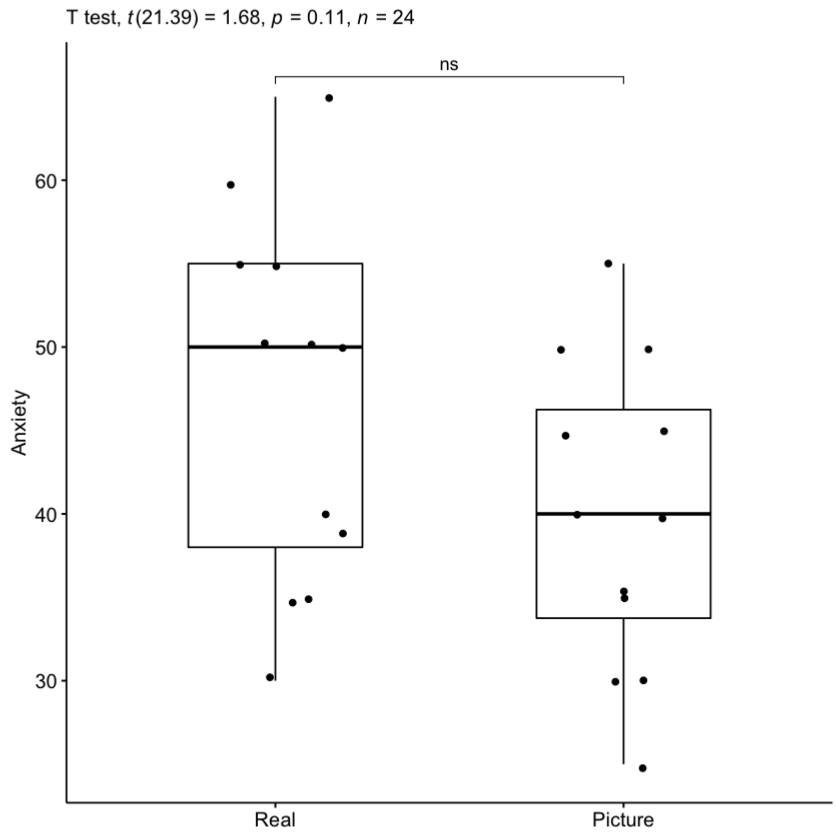
```
### [ Step.4 ] Effect size

Spider %>% cohens_d(Anxiety ~ Group)
```

A rstatix_test: 1 × 7

	.y.	group1	group2	effsize	n1	n2	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<int>	<ord>
1	Anxiety	Real	Picture	0.6864065	12	12	moderate

In [13]:

```
### [ Step.5 ] Visualization & Report

TwoSamp.test <- TwoSamp.test %>% add_xy_position(x="Group")

ggboxplot(Spider, x="Group", y="Anxiety", width = .5,
                      add = "jitter", ylab = "Anxiety", xlab = FALSE)+
    stat_pvalue_manual(TwoSamp.test, tip.length=0.01)+
    labs(subtitle = get_test_label(TwoSamp.test, detailed = TRUE))
```



~ *Report* ~

- **A two-sample t-test exhibited that the anxiety level of the real-spider group (47.0 ± 11.0) was NOT significantly higher than the anxiety level of the spider-picture group (40.0 ± 9.3) ($t_{21.4}$ = 1.68, *p* = .11, *d* = 4.1).**

- [Optional]: The anxiety levels of both groups were normally distributed (Shapiro-Wilk's test, $p > .05$) and fulfill the assumption of variance homogeneity (Levene's test, $p > .05$).

Example C: Spider Anxiety (as of 2 repeated measures)

[Hypothesis] Anxiety of viewing real spider is higher than anxiety of viewing spider picture. *(1-tailed)*

- Null hypothesis H_0 **:** Anxiety(real spider) ≤ Anxiety(spider picture)
- Alternative hyp. H_1 **:** Anxiety(real spider) > Anxiety(spider picture)

In [14]:

```
### [ Step.1 ] Data restructure (Long → Wide)

#----- Set up subject number -----#
Spider$Subj <- seq(1,12) %>% rep(2)

#----- Data restructure (Wide → Long) -----#
SpiderW <- Spider %>% spread(Group, Anxiety)
(SpiderW <- SpiderW %>% mutate(diff=Real-Picture))
```

A data.frame: 12 × 4

Subj	Real	Picture	diff
<int>	<int>	<int>	<int>
1	40	30	10
2	35	35	0
3	50	45	5
4	55	40	15
5	65	50	15
6	55	35	20
7	50	55	-5
8	35	25	10
9	30	30	0
10	50	45	5
11	60	40	20
12	39	50	-11

In [155]:

```
### [ Step.2- Step.4 ] Traditional Usage of t-test
# → not using {rstatix} package

pastecs::stat.desc(SpiderW$diff, basic = FALSE, desc = FALSE, norm = TRUE)

(Paired.test <- t.test(SpiderW$Real, SpiderW$Picture, paired = TRUE))

effsize::cohen.d(SpiderW$Real, SpiderW$Picture, paired=TRUE)
```

skewness: -0.246481020501118 **skew.2SE:** -0.193378506887084 **kurtosis:** -1.23421588806977 **kurt.2SE:** -0.500799115342087 **normtest.W:** 0.955790347989211 **normtest.p:** 0.72248006204431

```

	Paired t-test

data:  SpiderW$Real and SpiderW$Picture
t = 2.4725, df = 11, p-value = 0.03098
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  0.7687815 13.2312185
sample estimates:
mean of the differences 
                      7 


Cohen's d

d estimate: 0.6805173 (medium)
95 percent confidence interval:
     lower      upper 
0.04707723 1.31395743 
```

In [16]:

```
### [ Step.2- Step.4 ] Traditional Usage of t-test
# → using {rstatix} package

SpiderW %>% t_test(diff ~ 1, mu = 0)

SpiderW %>% cohens_d(diff ~ 1, mu = 0)
```

A rstatix_test: 1 × 7

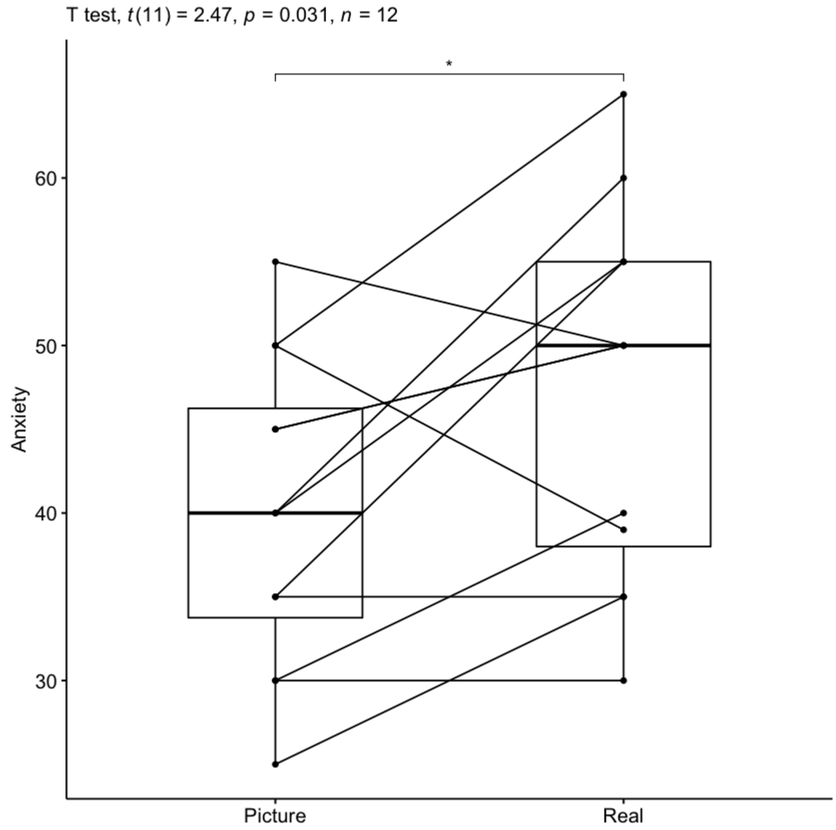
	.y.	group1	group2	n	statistic	df	p
	<chr>	<chr>	<chr>	<int>	<dbl>	<dbl>	<dbl>
1	diff	1	null model	12	2.472533	11	0.031

A rstatix_test: 1 × 6

	.y.	group1	group2	effsize	n	magnitude
	<chr>	<chr>	<chr>	<dbl>	<int>	<ord>
1	diff	1	null model	0.7137589	12	moderate

In [166]:

```
### [ Step.5 ] Visualization & Report

Paired.test <- Spider %>% t_test(Anxiety ~ Group, paired = TRUE) %>% add_significanc
Paired.test <- Paired.test %>% add_xy_position(x="Group")

ggpaired(Spider, x="Group", y="Anxiety", width = .5,
                 order = c("Picture", "Real"), ylab = "Anxiety", xlab = FALSE)+
    stat_pvalue_manual(Paired.test, tip.length=0.01, hide.ns = TRUE)+
    labs(subtitle = get_test_label(Paired.test, detailed = TRUE))
```



~ *Report* ~

- **A paired t-test exhibited that the anxiety level of watching real spider (47.0 ± 11.0) was higher than the anxiety level of watching spider picture (40.0 ± 9.3) of the same subjects (t_{11} = 2.47, *p* = .031, *d* = 0.68).**
- [Optional]: The difference of anxiety level was normally distributed (Shapiro-Wilk's test, $p > .05$).