8) Analysis of Variance (ANOVA)

Vitor Kamada

June 2018

Reference

Tables, Graphics, and Figures from

Introductory Statistics with Randomization and Simulation

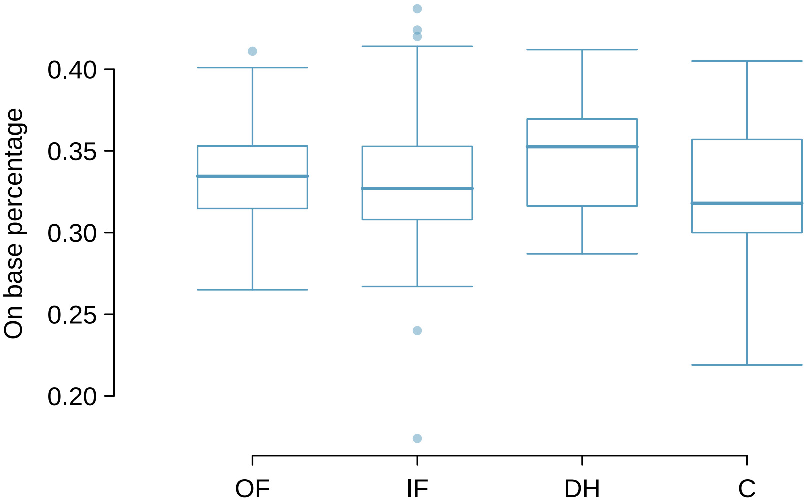
Diez et al. (2014): Chapter 4 - Inference for Numerical Data

Major League Baseball Data

	name	team	position	AB	H	HR	RBI	AVG	OBP
1	I Suzuki	SEA	OF	680	214	6	43	0.315	0.359
2	D Jeter	NYY	IF	663	179	10	67	0.270	0.340
3	M Young	TEX	IF	656	186	21	91	0.284	0.330
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		
325	B Molina	SF	C	202	52	3	17	0.257	0.312
326	J Thole	NYM	C	202	56	3	17	0.277	0.357
327	C Heisey	CIN	OF	201	51	8	21	0.254	0.324

variable	**description**
position	The player’s primary field position (OF, IF, DH, C)
AB	Number of opportunities at bat
H	Number of hits
HR	Number of home runs
RBI	Number of runs batted in
AVG	Batting average, which is equal to H/AB
OBP	On-base percentage, which is roughly equal to the fraction of times a player gets on base or hits a home run

Is Batting Performance related to Player Position?



Summary Statistics of On-base Percentage

$$H_0 : \mu_{OF} = \mu_{IF} = \mu_{DH} = \mu_C$$

H_A : The average on-base percentage (μ_i) varies across some (or all) groups.

	OF	IF	DH	C
Sample size (n_i)	120	154	14	39
Sample mean ($\bar{x}_i$)	0.334	0.332	0.348	0.323
Sample SD (s_i)	0.029	0.037	0.036	0.045

F-test

Mean Square between Groups:

$$MSG = \frac{SSG}{df_G} = \frac{1}{k-1} \sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})^2$$

Mean Square Error:

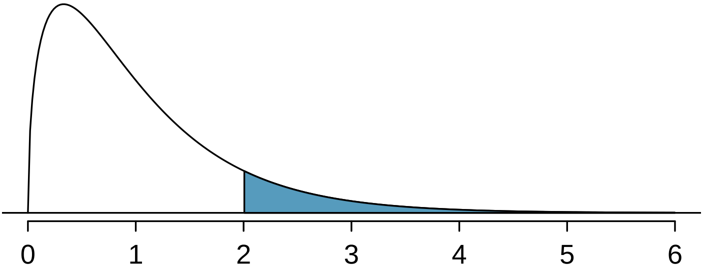
$$MSE = \frac{SSE}{df_E} = \frac{1}{n-k} \sum_{i=1}^{k} (n_i - 1)s_i^2$$

$$F = \frac{MSG}{MSE}$$

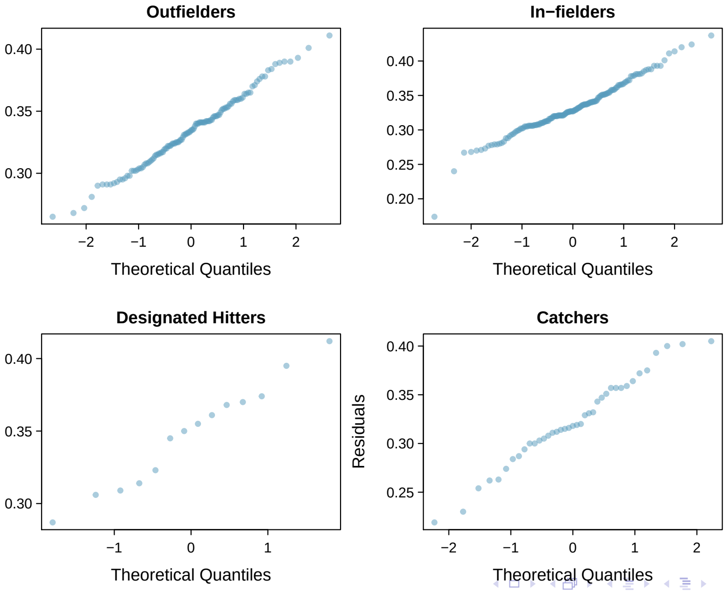
Analysis of Variance (ANOVA)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
position	3	0.0076	0.0025	1.9943	0.1147
Residuals	323	0.4080	0.0013		

$s_{pooled} = 0.036$ on $df = 323$



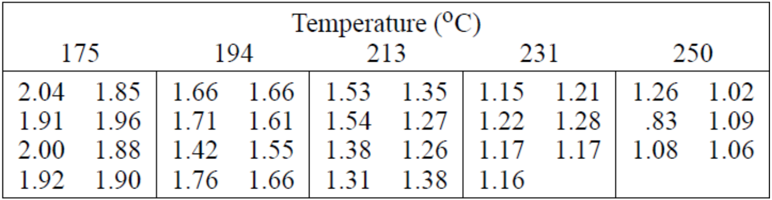
Normal Probability Plot of On-base Percentage



Log(Lifetime) of Resin in Integrated Circuits

```
import pandas as pd

resin =
pd.read_table('http://www.stat.umn.edu//~gary//book//fcdae.data//exmp
header=10, delim_whitespace=True)
```



Temperature (°C)									
175		194		213		231		250	
2.04	1.85	1.66	1.66	1.53	1.35	1.15	1.21	1.26	1.02
1.91	1.96	1.71	1.61	1.54	1.27	1.22	1.28	.83	1.09
2.00	1.88	1.42	1.55	1.38	1.26	1.17	1.17	1.08	1.06
1.92	1.90	1.76	1.66	1.31	1.38	1.16			

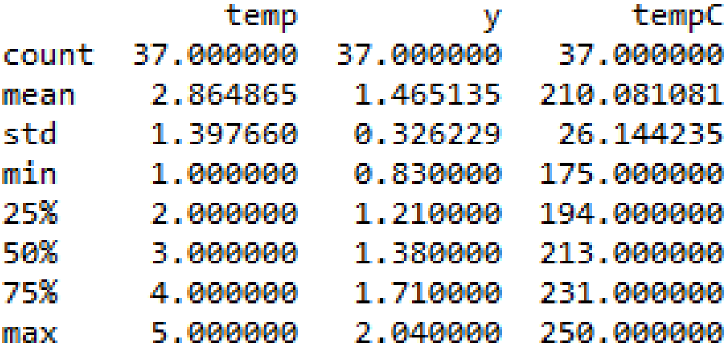
Nelson (1990)

Recode Variable

```
def temp_groups(series):
    if series == 1:
        return 175
    elif series == 2:
        return 194
    elif series == 3:
        return 213
    elif series == 4:
        return 231
    elif series == 5:
        return 250
resin['tempC'] = resin['temp'].apply(temp_groups)
```

Descriptive Statistics

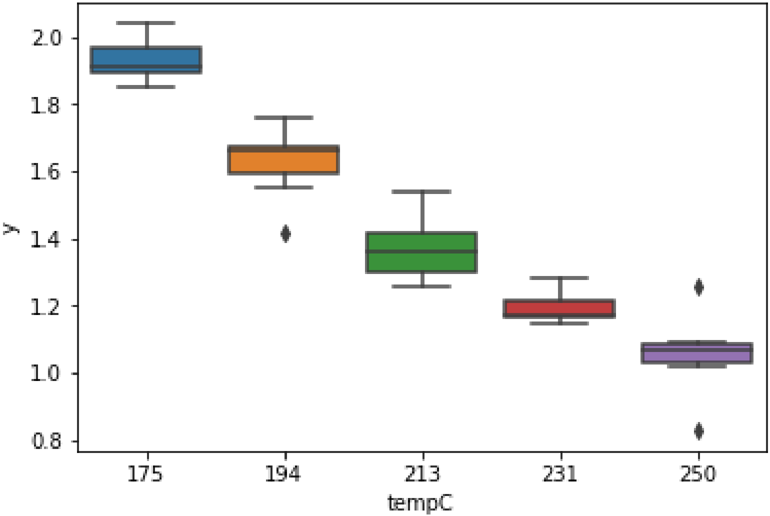
resin.describe()



```
            temp          y       tempC
count  37.000000  37.000000   37.000000
mean    2.864865   1.465135  210.081081
std     1.397660   0.326229   26.144235
min     1.000000   0.830000  175.000000
25%     2.000000   1.210000  194.000000
50%     3.000000   1.380000  213.000000
75%     4.000000   1.710000  231.000000
max     5.000000   2.040000  250.000000
```

import seaborn as sns

```
sns.boxplot(x="tempC", y="y", data=resin)
```



ANOVA Table

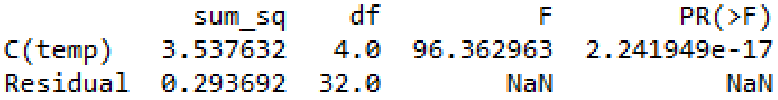
```
import statsmodels.api as sm

from statsmodels.formula.api import ols

Model = ols(’y ∼ C(temp)’, data=resin).fit()

table = sm.stats.anova_lm(Model, typ=2)

print(table)
```



```
             sum_sq    df          F        PR(>F)
C(temp)    3.537632   4.0  96.362963  2.241949e-17
Residual   0.293692  32.0        NaN           NaN
```

F Distribution

```
import scipy.stats

scipy.stats.f.cdf(96.363, 4, 32)
```

1

```
1-scipy.stats.f.cdf(96.363, 4, 32)
```

0

```
scipy.stats.f.ppf(0.999999999999, 4, 32)
```

86.38