An Incomplete Survey of Fundamental Statistical Tests in R

EDP 613: Fall 2020

Libraries

Please load up the following packages (or install and then load them as needed)

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
```

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

```
library(foreign)
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(CCA)
```

```
## Loading required package: fda

## Loading required package: splines

## Loading required package: fds

## Loading required package: rainbow
```

```
## Loading required package: pcaPP

## Loading required package: RCurl

##
## Attaching package: 'RCurl'

## The following object is masked from 'package:tidyr':
##
##     complete

## Loading required package: deSolve

##
## Attaching package: 'fda'

## The following object is masked from 'package:graphics':
##
##     matplot

## Loading required package: fields

## Loading required package: spam

## Loading required package: dotCall64

## Loading required package: grid

## Spam version 2.7-0 (2021-06-25) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following object is masked from 'package:Matrix':
##
##     det

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:fields':
##
##     describe

## The following object is masked from 'package:car':
##
##     logit
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

A Side Note About R

Fist of all, a big thank you for sticking with R even though it was rough at times. It may not be apparent, but developing coding skills like the ones in this course have benefits, not least of all in simply understanding the structure of a given data set. There are too many examples of students and even professionals who run an analysis on data without considering the data itself. R and other syntax-based software packages like it (yes even SPSS if you work with the underlying code) to their credit make you explore your data whether it be through checks or frustration.

While I agree that proprietary softwares such as SPSS, SAS, Minitab, etc. are easier to learn R (and Python) is open access (free) while proprietary softwares requires an expensive license. Picking on SPSS, users pay for the base package and then have to pay separately for any add-ons (such as predictive analytics, generate data from missingness, etc.) they want to use. Consider what would make you most marketable!

With that said, learning R is a lifelong process and assisting student learning and growth should never be confined to a single course so please FEEL FREE to contact me if you have questions regarding R (or Python if you go there) at any time. Again, I will always make time for students.

Purpose

This walk-through will provide you with information on how to perform a number of statistical tests using R. Some of these will look familiar while others you will be exposed to in future statistics courses if that is your path. In either case, hopefully these will be helpful if for no other reason than to provide a check or confirmation of results.

Decisions Decisions Decisions

When deciding which test is appropriate to use, it is important to consider the type of variables that you have. Please load in the following data sets (AND TAKE A LOOK AT THEM by using View() or head())

```
some_ed_data <- read_csv("some_ed_data.csv")
```

```
## Rows: 200 Columns: 11
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (11): id, female, race, ses, schtyp, prog, read, write, math, science, s...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
some_exercise_data <- read_csv("some_exercise_data.csv")
```

```
## Rows: 90 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): id, diet, exertype, pulse, time, highpulse
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
some_survey_data <- read_csv("some_survey_data.csv")
```

```
## Rows: 20 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (4): respondent, gender, item_1, item_2
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

The following table provides some basic information about data types. While I would love to have constructed this table and the syntax blocks thereafter, a majority of the information was scraped from the web using R via the UCLA Institute for Digital Research & Education site using the xml2 package. They also fully support SAS, SPSS (for those of you moving on to EDP 614), Stata, and Mplus.

NOTE: I do not not receive compensation of any kind from the organization. With that said, it is an excellent resource so you may want to consider using it as a point of reference.

The (Not So) Ultimate Table

Number of Dependent Variables	Type of Independent Variables	Type of Dependent Variables	Test(s)
1	0 IVs (1 population)	interval & normal	one-sample t-test
1	0 IVs (1 population)	ordinal or interval	one-sample median
1	0 IVs (1 population)	categorical (2 categories)	binomial test
1	0 IVs (1 population)	categorical	Chi-square goodness-of-fit
1	1 IV with 2 levels (independent groups)	interval & normal	2 independent sample t-test
1	1 IV with 2 levels (independent groups)	ordinal or interval	Wilcoxon-Mann Whitney test
1	1 IV with 2 levels (independent groups)	categorical	Chi-square test
1	1 IV with 2 levels (independent groups)	categorical	Fisher's exact test
1	1 IV with 2 or more levels (independent groups)	interval & normal	one-way ANOVA
1	1 IV with 2 or more levels (independent groups)	ordinal or interval	Kruskal Wallis
1	1 IV with 2 or more levels (independent groups)	categorical	Chi-square test
1	1 IV with 2 levels (dependent/matched groups)	interval & normal	paired t-test
1	1 IV with 2 levels (dependent/matched groups)	ordinal or interval	Wilcoxon signed ranks test
1	1 IV with 2 levels (dependent/matched groups)	categorical	McNemar
1	1 IV with 2 or more levels (dependent/matched groups)	interval & normal	one-way repeated measures ANOVA
1	1 IV with 2 or more levels (dependent/matched groups)	ordinal or interval	Friedman test
1	1 IV with 2 or more levels (dependent/matched groups)	categorical (2 categories)	repeated measures logistic regression
1	2 or more IVs (independent groups)	interval & normal	factorial ANOVA
1	2 or more IVs (independent groups)	ordinal or interval	ordered logistic regression
1	2 or more IVs (independent groups)	categorical (2 categories)	factorial logistic regression
1	1 interval IV	interval & normal	correlation
1	1 interval IV	interval & normal	simple linear regression
1	1 interval IV	ordinal or interval	non-parametric correlation
1	1 interval IV	categorical	simple logistic regression
1	1 or more interval IVs and/or 1 or more categorical IVs	interval & normal	multiple regression
1	1 or more interval IVs and/or 1 or more categorical IVs	interval & normal	analysis of covariance
1	1 or more interval IVs and/or 1 or more categorical IVs	categorical	multiple logistic regression
1	1 or more interval IVs and/or 1 or more categorical IVs	categorical	discriminant analysis
2+	1 IV with 2 or more levels (independent groups)	interval & normal	one-way MANOVA
2+	2+	interval & normal	multivariate multiple linear

Tests

ANCOVA (Analysis of Covariance)

```
summary(aov(some_ed_data$write ~ some_ed_data$prog + some_ed_data$read))
```

```
##                    Df Sum Sq Mean Sq F value  Pr(>F)
## some_ed_data$prog   1    586     586    10.2 0.00164 **
## some_ed_data$read   1   5965    5965   103.7 < 2e-16 ***
## Residuals         197  11327      57
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Binomial test

```
prop.test(sum(some_ed_data$female), length(some_ed_data$female), p = 0.5)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  sum(some_ed_data$female) out of length(some_ed_data$female), null probability 0.5
## X-squared = 1.445, df = 1, p-value = 0.2293
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.4733037 0.6149394
## sample estimates:
##     p
## 0.545
```

Canonical Correlation

```
cc(cbind(some_ed_data$read, some_ed_data$write), cbind(some_ed_data$math,
                                                        some_ed_data$science))
```

```
## $cor
## [1] 0.7728409 0.0234784
##
## $names
## $names$Xnames
## NULL
##
## $names$Ynames
## NULL
##
## $names$ind.names
## NULL
##
##
## $xcoef
##             [,1]       [,2]
## [1,] -0.06326131 -0.1037908
## [2,] -0.04924918  0.1219084
##
## $ycoef
```

```
##              [,1]       [,2]
## [1,] -0.06698268  0.1201425
## [2,] -0.04824063 -0.1208860
##
## $scores
## $scores$xscores
##               [,1]         [,2]
##   [1,] -0.26358835 -0.589561062
##   [2,] -1.30420707 -0.877901269
##   [3,]  1.49454321 -1.556539586
##   [4,] -0.24916276 -2.187572699
##   [5,]  0.36902478  0.448346869
##   [6,]  0.55880872  0.759719249
##   [7,] -0.16550344  0.990333008
##   [8,]  1.48691695  1.066177022
##   [9,] -0.88940214 -0.602764022
##  [10,] -0.41133590 -0.223835983
##  [11,] -0.15787718 -1.632383600
##  [12,] -0.90382773  0.995247615
##  [13,] -1.66976282 -1.274946875
##  [14,] -0.61554542  1.062803275
##  [15,]  0.24930149  1.265470254
##  [16,]  0.83307890  0.601575756
##  [17,]  0.36902478  0.448346869
##  [18,] -0.50983426  0.019980737
##  [19,] -1.59970217 -0.146451110
##  [20,]  0.50317366 -1.966788153
##  [21,] -0.49540867 -1.578030900
##  [22,] -1.18489724  0.128686136
##  [23,]  0.77023105 -1.325925827
##  [24,] -0.45337228 -0.900933442
##  [25,]  1.39563138  0.510987923
##  [26,]  1.93015961 -0.030998216
##  [27,] -1.40991823  0.164921269
##  [28,]  0.12277887  1.057888668
##  [29,]  1.24829729 -0.946997788
##  [30,]  0.44671169 -1.045873974
##  [31,]  1.71956420 -1.592774720
##  [32,] -1.46555329 -2.561586132
##  [33,] -1.50841660  0.408737989
##  [34,]  1.22707237 -0.373691122
##  [35,] -0.29202606  0.782751422
##  [36,] -1.09361167  0.683875235
##  [37,] -1.05796116 -1.487443067
##  [38,] -1.26217068 -0.200803810
##  [39,]  1.40325764 -2.111728685
##  [40,]  1.90934814 -1.281402340
##  [41,]  0.61527070 -0.161194929
##  [42,] -0.48181000  0.471379042
##  [43,] -0.04578015  0.173209623
##  [44,]  1.22707237 -0.373691122
##  [45,] -1.40991823  0.164921269
##  [46,] -0.48181000  0.471379042
##  [47,]  0.77023105 -1.325925827
```

```
##  [48,] -1.11442313 -0.566528889
##  [49,]  0.02428050  1.301705388
##  [50,] -0.36208671 -0.345744343
##  [51,] -0.45378574  0.922777348
##  [52,]  1.81084977 -1.037585621
##  [53,]  0.95280219 -0.215547629
##  [54,] -0.98790051 -0.358947303
##  [55,] -1.76826119 -1.031130155
##  [56,]  1.51535467 -0.306135462
##  [57,]  1.74037567 -0.342370595
##  [58,]  1.32557073 -0.617507842
##  [59,] -0.88940214 -0.602764022
##  [60,]  0.45351102 -0.021169003
##  [61,]  0.47473595 -0.594475669
##  [62,] -0.12346705  1.667430467
##  [63,]  0.95280219 -0.215547629
##  [64,]  2.12715634 -0.518631655
##  [65,]  0.67173268 -1.082109108
##  [66,]  1.10054974 -0.581272708
##  [67,] -0.55187065 -0.657116722
##  [68,]  1.00926417 -1.136461807
##  [69,] -0.19991357 -2.309481059
##  [70,]  1.11497533 -2.179284345
##  [71,]  0.95280219 -0.215547629
##  [72,] -0.55187065 -0.657116722
##  [73,] -2.01450711 -0.421588356
##  [74,] -1.30420707 -0.877901269
##  [75,]  0.20767856 -1.235337994
##  [76,]  0.96001499 -1.014553448
##  [77,] -0.58030837  0.715195762
##  [78,] -1.30420707 -0.877901269
##  [79,]  2.16919273  0.158465804
##  [80,]  0.91076580 -0.892645088
##  [81,] -0.98790051 -0.358947303
##  [82,]  1.21264678  1.224320515
##  [83,] -1.30420707 -0.877901269
##  [84,] -1.28339561  0.372502856
##  [85,]  0.40467530 -1.722971433
##  [86,] -0.62955755  0.837104122
##  [87,]  0.59445924 -1.411599054
##  [88,]  1.33916940  1.431902101
##  [89,]  1.21347370 -2.423101065
##  [90,]  0.01068183 -0.747704555
##  [91,] -0.43977362  1.148476501
##  [92,] -0.49540867 -1.578030900
##  [93,] -1.45195462 -0.512176189
##  [94,]  1.26910876  0.303406337
##  [95,]  0.95280219 -0.215547629
##  [96,] -0.31325099  1.356058087
##  [97,] -1.78948611 -0.457823489
##  [98,] -1.28339561  0.372502856
##  [99,]  1.58541532  0.822360302
## [100,] -1.18489724  0.128686136
## [101,] -1.35345626 -0.755992909
```

```
## [102,]  0.02428050  1.301705388
## [103,]  0.66451988 -0.283103289
## [104,] -0.64315622 -1.212305821
## [105,] -0.29202606  0.782751422
## [106,] -0.23556409 -0.138162756
## [107,] -0.94586412  0.318150156
## [108,]  1.96539666 -0.378605729
## [109,]  0.27052642  0.692163589
## [110,] -1.78948611 -0.457823489
## [111,] -0.26358835 -0.589561062
## [112,]  0.65730709  0.515902529
## [113,] -1.11442313 -0.566528889
## [114,] -1.59970217 -0.146451110
## [115,] -1.71901201 -1.153038515
## [116,]  1.45889269  0.614778716
## [117,]  0.52357167  1.107326761
## [118,] -2.01450711 -0.421588356
## [119,] -0.19352770  0.538934702
## [120,]  0.99483858  0.461549830
## [121,] -0.55187065 -0.657116722
## [122,]  0.17924085  0.136974490
## [123,]  0.17924085  0.136974490
## [124,]  0.66451988 -0.283103289
## [125,] -0.12346705  1.667430467
## [126,] -0.38331164  0.227562323
## [127,]  0.72098186 -1.204017467
## [128,]  0.82586610  1.400581574
## [129,]  0.32698840 -0.228750589
## [130,]  2.02865797 -0.274814935
## [131,] -0.60833263  0.263797456
## [132,] -0.90382773  0.995247615
## [133,] -1.45195462 -0.512176189
## [134,]  0.58683298  1.211117555
## [135,] -0.86137788 -0.151365716
## [136,] -2.00729431 -1.220594175
## [137,]  0.02428050  1.301705388
## [138,]  0.43228610  0.552137663
## [139,]  1.41685631 -0.062318743
## [140,]  0.20046577 -0.436332176
## [141,]  1.86648483  1.688921781
## [142,]  0.58683298  1.211117555
## [143,]  0.86151662 -0.770736728
## [144,]  0.12277887  1.057888668
## [145,] -0.29202606  0.782751422
## [146,]  0.36902478  0.448346869
## [147,] -0.31325099  1.356058087
## [148,]  0.55880872  0.759719249
## [149,]  0.91076580 -0.892645088
## [150,]  0.34779986  1.021653535
## [151,]  1.10776254 -1.380278527
## [152,] -0.86817722 -1.176070688
## [153,]  0.37582412  1.473051841
## [154,]  0.27052642  0.692163589
## [155,] -0.75608018  0.629522535
```

```
## [156,] -1.30420707 -0.877901269
## [157,] -0.09502933  0.295117983
## [158,]  0.43908543  1.576842634
## [159,]  1.32557073 -0.617507842
## [160,] -1.57167791  0.304947196
## [161,] -0.12346705  1.667430467
## [162,] -0.16508998 -0.833377782
## [163,] -0.07421787  1.545522107
## [164,] -0.75608018  0.629522535
## [165,] -0.29202606  0.782751422
## [166,]  0.95280219 -0.215547629
## [167,] -0.16550344  0.990333008
## [168,]  0.77661692  1.522489934
## [169,] -0.75608018  0.629522535
## [170,] -0.65758181  0.385705816
## [171,]  0.43908543  1.576842634
## [172,]  0.66451988 -0.283103289
## [173,]  1.47331828 -0.983232921
## [174,] -0.79811657 -0.047574923
## [175,]  0.70655627  0.393994170
## [176,] -1.03714969 -0.237038943
## [177,] -1.50841660  0.408737989
## [178,]  0.77661692  1.522489934
## [179,]  0.17924085  0.136974490
## [180,] -0.58752117  1.514201580
## [181,] -0.94586412  0.318150156
## [182,]  0.70655627  0.393994170
## [183,] -0.68601953  1.758018300
## [184,] -0.77730510  1.202829201
## [185,] -0.55949691  1.965599886
## [186,] -1.40991823  0.164921269
## [187,] -0.04578015  0.173209623
## [188,]  0.76301825 -0.526920009
## [189,] -1.13564806  0.006777776
## [190,]  0.47473595 -0.594475669
## [191,]  0.58683298  1.211117555
## [192,]  0.81865331  2.199587393
## [193,]  0.17924085  0.136974490
## [194,]  0.40384838  1.924450147
## [195,] -0.27121460  2.033155546
## [196,] -0.48181000  0.471379042
## [197,]  0.98082645  0.235850677
## [198,]  0.27815267 -1.930553019
## [199,] -0.62955755  0.837104122
## [200,] -1.28339561  0.372502856
##
## $scores$yscores
##                [,1]        [,2]
##    [1,]  1.013980334 -0.81276184
##    [2,] -0.561661891 -1.30522812
##    [3,] -0.387441410 -0.58065576
##    [4,]  0.322640484 -0.81722302
##    [5,] -0.347186284  0.38420150
##    [6,] -0.427696537 -1.54551302
```

```
##   [7,]  0.657553868 -1.41793527
##   [8,]  1.131974678  0.63489582
##   [9,] -0.387441410 -0.58065576
##  [10,]  0.132448995  0.14614718
##  [11,]  0.054709777 -0.33665321
##  [12,] -0.427696537 -1.54551302
##  [13,] -1.670868810  1.09910797
##  [14,] -0.443667546  0.14242953
##  [15,]  1.182986345  2.20269592
##  [16,]  0.735293086 -0.93513488
##  [17,]  0.199431671  0.02600473
##  [18,] -0.789337471  0.14019895
##  [19,] -0.778580931  0.74314179
##  [20,] -0.347186284  0.38420150
##  [21,]  0.499304397 -3.83045019
##  [22,] -2.469446463  0.24993198
##  [23,]  0.360124575 -1.29927988
##  [24,] -0.733111335 -0.58288635
##  [25,]  1.131974678  0.63489582
##  [26,]  1.064992001  0.75503827
##  [27,] -1.335955426  0.49839571
##  [28,] -0.588389441 -0.22022841
##  [29,]  0.864043971  1.11546562
##  [30,]  0.092193868 -0.81871008
##  [31,] -0.057742495  1.10951738
##  [32,] -1.346711967 -0.10454714
##  [33,] -1.376210553 -0.46646155
##  [34,] -1.263758281 -1.91263214
##  [35,] -0.264232597 -1.42388351
##  [36,] -0.800094012 -0.46274390
##  [37,] -2.180002675  0.97524787
##  [38,] -1.711123937  0.13425071
##  [39,]  1.343679249  0.87741131
##  [40,]  1.466888062  0.03418356
##  [41,]  1.255183491 -0.20833193
##  [42,] -0.923302825  0.38048385
##  [43,] -0.443667546  0.14242953
##  [44,]  0.735293086 -0.93513488
##  [45,] -0.226748507 -1.90594038
##  [46,] -1.520932447 -0.82911949
##  [47,]  1.051464425 -1.29481870
##  [48,] -1.700367396  0.73719355
##  [49,] -0.749082344  1.10505620
##  [50,] -0.191707849  1.34980229
##  [51,] -0.808079517  0.38122738
##  [52,]  1.214928364 -1.17318919
##  [53,] -0.470395097  1.22742925
##  [54,]  0.092193868 -0.81871008
##  [55,] -2.190759215  0.37230502
##  [56,]  1.826085563  2.08627112
##  [57,]  1.622366497  0.99978435
##  [58,]  0.713780005 -2.14102057
##  [59,] -0.454424087 -0.46051331
##  [60,]  0.421892783  0.87146307
```

```
##   [61,]  0.215402681 -1.66193782
##   [62,] -1.386967094 -1.06940440
##   [63,]  1.775073896  0.51847102
##   [64,]  1.544627280  0.51698396
##   [65,]  0.941783188  1.59826602
##   [66,]  0.936241116 -1.29556223
##   [67,] -0.644615577  0.50285689
##   [68,]  0.853287430  0.51252278
##   [69,] -1.060039214 -0.82614537
##   [70,]  0.622840814  0.51103572
##   [71,]  1.064992001  0.75503827
##   [72,] -0.655372118 -0.10008596
##   [73,] -1.767350073  0.85733600
##   [74,] -1.427222220 -2.03426166
##   [75,]  0.148420004 -1.54179537
##   [76,]  0.976496243 -0.33070497
##   [77,]  0.046724273  0.50731807
##   [78,] -0.762609921 -0.94480077
##   [79,]  1.064992001  0.75503827
##   [80,]  0.384408693  1.35351994
##   [81,] -0.443667546  0.14242953
##   [82,] -0.532163305 -0.94331371
##   [83,] -1.912071967  0.49467806
##   [84,] -0.226748507 -1.90594038
##   [85,]  1.225684905 -0.57024634
##   [86,] -1.298471336  0.01633884
##   [87,]  0.054709777 -0.33665321
##   [88,]  1.389148845 -0.44861683
##   [89,]  1.708091219  0.63861347
##   [90,] -1.001042042 -0.10231655
##   [91,] -0.660586586  2.19079945
##   [92,] -0.438453078 -2.14845587
##   [93,] -1.654897801 -0.58883459
##   [94,]  0.421892783  0.87146307
##   [95,]  0.679066950 -0.21204958
##   [96,] -1.097523305 -0.34408851
##   [97,] -1.979054644  0.61482051
##   [98,] -2.056793862  0.13202012
##   [99,]  1.466888062  0.03418356
##  [100,] -1.536903457  0.85882306
##  [101,] -1.587915124 -0.70897704
##  [102,] -0.907331815 -1.30745871
##  [103,]  1.153487759  1.84078151
##  [104,] -0.001516359  0.38643209
##  [105,] -0.465180628 -1.06345616
##  [106,] -0.419383429  2.79522935
##  [107,] -0.872291157  1.94828395
##  [108,]  0.888000485 -1.41644821
##  [109,]  0.534345056 -0.57470752
##  [110,] -1.392181562  1.22148101
##  [111,]  0.405594171 -2.62530802
##  [112,]  1.399905385  0.15432601
##  [113,] -0.009501864  1.23040337
##  [114,] -0.703612749 -0.22097194
```

```
## [115,] -0.443667546  0.14242953
## [116,]  1.523114198 -0.68890174
## [117,] -0.309702193 -0.09785537
## [118,] -0.923302825  0.38048385
## [119,] -1.057268178  0.62076875
## [120,]  1.466888062  0.03418356
## [121,]  0.266414348 -0.09413772
## [122,]  0.534345056 -0.57470752
## [123,]  0.596113264  1.59603543
## [124,]  0.228930258  0.38791915
## [125,]  1.373177835  1.23932572
## [126,] -0.521406764 -0.34037086
## [127,]  0.890771521  0.03046591
## [128,] -0.001516359  0.38643209
## [129,]  0.009240182  0.98937493
## [130,]  1.882311700  1.36318582
## [131,] -0.001516359  0.38643209
## [132,] -1.400167067  2.06545229
## [133,] -0.135481713  0.62671699
## [134,]  0.612084273 -0.09190713
## [135,]  0.622840814  0.51103572
## [136,] -1.223503154 -0.94777488
## [137,] -0.387441410 -0.58065576
## [138,]  0.220944753  1.23189042
## [139,]  1.791044905 -1.16947154
## [140,]  0.622840814  0.51103572
## [141,]  1.024736875 -0.20981899
## [142,]  0.266414348 -0.09413772
## [143,]  0.663095940  1.47589298
## [144,]  0.689823490  0.39089327
## [145,] -0.414168960  0.50434395
## [146,]  0.831774349 -0.69336292
## [147,]  0.628055282 -1.77984969
## [148,]  1.024736875 -0.20981899
## [149,]  1.016751370  0.63415229
## [150,]  1.440160512  1.11918327
## [151,]  1.121218137  0.03195297
## [152,] -0.856320148  0.26034140
## [153,]  0.936241116 -1.29556223
## [154,]  0.188675131 -0.57693811
## [155,] -0.521406764 -0.34037086
## [156,] -0.711598254  0.62299934
## [157,]  0.073451823 -0.57768164
## [158,]  0.344153566  0.38866268
## [159,]  1.188200814 -0.08818948
## [160,] -1.402938103  0.61853816
## [161,] -0.079255576 -0.09636831
## [162,]  0.218173717 -0.21502370
## [163,] -0.427696537 -1.54551302
## [164,] -1.737851487  1.21925042
## [165,]  0.159176545 -0.93885253
## [166,]  1.418647431 -0.08670242
## [167,] -0.465180628 -1.06345616
## [168,]  1.147945687 -1.05304674
```

```
## [169,] -0.845563607  0.86328424
## [170,]  0.054709777 -0.33665321
## [171,]  0.601327732 -0.69484998
## [172,]  1.147945687 -1.05304674
## [173,]  1.718847760  1.24155631
## [174,] -1.068024719  0.01782590
## [175,]  1.391919881  0.99829729
## [176,] -0.931288329  1.22445513
## [177,] -0.845563607  0.86328424
## [178,] -0.146238253  0.02377414
## [179,]  0.215402681 -1.66193782
## [180,] -0.692856208  0.38197091
## [181,]  0.333397025 -0.21428017
## [182,]  0.931026648  0.99532317
## [183,] -0.829592598 -0.82465831
## [184,] -0.068499036  0.50657454
## [185,] -1.577158583 -0.10603420
## [186,] -1.057268178  0.62076875
## [187,] -0.213220930  0.14391659
## [188,]  1.188200814 -0.08818948
## [189,] -0.376684870  0.02228708
## [190,] -0.079255576 -0.09636831
## [191,]  1.255183491 -0.20833193
## [192,]  0.802275763 -1.05527733
## [193,] -0.175736839 -0.33814027
## [194,]  1.574125866  0.87889837
## [195,] -0.403412420  1.10728679
## [196,]  0.518374046  1.11323503
## [197,]  1.745575310  0.15655660
## [198,] -0.443667546  0.14242953
## [199,] -0.655372118 -0.10008596
## [200,] -0.883047698  1.34534111
##
## $scores$corr.X.xscores
##            [,1]      [,2]
## [1,] -0.9271970 -0.374574
## [2,] -0.8538903  0.520453
##
## $scores$corr.Y.xscores
##            [,1]         [,2]
## [1,] -0.7177974  0.008701966
## [2,] -0.6750187 -0.011433002
##
## $scores$corr.X.yscores
##            [,1]         [,2]
## [1,] -0.7165758 -0.008794398
## [2,] -0.6599214  0.012219404
##
## $scores$corr.Y.yscores
##            [,1]       [,2]
## [1,] -0.9287778  0.3706371
## [2,] -0.8734252 -0.4869583
```

Chi-square (χ^2) test

```r
chisq.test(table(some_ed_data$female, some_ed_data$schtyp))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(some_ed_data$female, some_ed_data$schtyp)
## X-squared = 0.00054009, df = 1, p-value = 0.9815
```

Chi-square (χ^2) Goodness of Fit

```r
chisq.test(table(some_ed_data$race), p = c(10, 10, 10, 70)/100)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(some_ed_data$race)
## X-squared = 5.0286, df = 3, p-value = 0.1697
```

Correlation

```r
cor(some_ed_data$read, some_ed_data$write)
```

```
## [1] 0.5967765
```

```r
cor.test(some_ed_data$read, some_ed_data$write)
```

```
##
##  Pearson's product-moment correlation
##
## data:  some_ed_data$read and some_ed_data$write
## t = 10.465, df = 198, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4993831 0.6792753
## sample estimates:
##       cor
## 0.5967765
```

Discriminant analysis

```r
fit <- lda(factor(some_ed_data$prog) ~ some_ed_data$read +
             some_ed_data$write + some_ed_data$math, data = some_ed_data)

fit
```

```
## Call:
## lda(factor(some_ed_data$prog) ~ some_ed_data$read + some_ed_data$write +
##     some_ed_data$math, data = some_ed_data)
##
## Prior probabilities of groups:
##     1     2     3
## 0.225 0.525 0.250
```

```
##
## Group means:
##   some_ed_data$read some_ed_data$write some_ed_data$math
## 1          49.75556           51.33333          50.02222
## 2          56.16190           56.25714          56.73333
## 3          46.20000           46.76000          46.42000
##
## Coefficients of linear discriminants:
##                          LD1         LD2
## some_ed_data$read  0.02919876  0.04385321
## some_ed_data$write 0.03832289 -0.13698224
## some_ed_data$math  0.07034625  0.07931008
##
## Proportion of trace:
##    LD1    LD2
## 0.9874 0.0126
```

Factor analysis

```
fa(r = cor(model.matrix(~read + write + math + science + socst - 1,
                        data = some_ed_data)), rotate = "none", fm = "pa", 2)
```

```
## maximum iteration exceeded

## Factor Analysis using method =  pa
## Call: fa(r = cor(model.matrix(~read + write + math + science + socst -
##     1, data = some_ed_data)), nfactors = 2, rotate = "none",
##     fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##          PA1   PA2   h2   u2 com
## read    0.81  0.06 0.66 0.34 1.0
## write   0.76  0.00 0.58 0.42 1.0
## math    0.80  0.17 0.67 0.33 1.1
## science 0.75  0.26 0.62 0.38 1.2
## socst   0.79 -0.48 0.85 0.15 1.6
##
##                        PA1  PA2
## SS loadings           3.06 0.33
## Proportion Var        0.61 0.07
## Cumulative Var        0.61 0.68
## Proportion Explained  0.90 0.10
## Cumulative Proportion 0.90 1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are  10  and the objective function was  2.51
## The degrees of freedom for the model are 1  and the objective function was  0.01
##
## The root mean square of the residuals (RMSR) is  0.01
## The df corrected root mean square of the residuals is  0.03
##
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
```

```
##                                                  PA1  PA2
## Correlation of (regression) scores with factors   0.95 0.79
## Multiple R square of scores with factors          0.91 0.62
## Minimum correlation of possible factor scores     0.82 0.23
```

Factorial ANOVA (Analysis of Variance)

```
anova(lm(write ~ female * ses, data = some_ed_data))
```

```
## Analysis of Variance Table
##
## Response: write
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## female        1  1176.2 1176.21 14.7212 0.0001680 ***
## ses           1  1042.3 1042.32 13.0454 0.0003862 ***
## female:ses    1     0.0    0.04  0.0005 0.9827570
## Residuals   196 15660.3   79.90
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Factorial Logistic Regression

```
summary(glm(female ~ prog * schtyp, data = some_ed_data, family = binomial))
```

```
##
## Call:
## glm(formula = female ~ prog * schtyp, family = binomial, data = some_ed_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.698  -1.247    1.069    1.109    1.572
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.2765     1.8857  -1.207    0.227
## prog          1.2303     0.9398   1.309    0.191
## schtyp        2.2405     1.7017   1.317    0.188
## prog:schtyp  -1.1313     0.8622  -1.312    0.189
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 275.64  on 199  degrees of freedom
## Residual deviance: 273.65  on 196  degrees of freedom
## AIC: 281.65
##
## Number of Fisher Scoring iterations: 4
```

Friedman test

```
friedman.test(cbind(some_ed_data$read, some_ed_data$write, some_ed_data$math))
```

```
##
##  Friedman rank sum test
```

```
##
## data:  cbind(some_ed_data$read, some_ed_data$write, some_ed_data$math)
## Friedman chi-squared = 0.64491, df = 2, p-value = 0.7244
```

Kruskal Wallis test

```r
kruskal.test(some_ed_data$write, some_ed_data$prog)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  some_ed_data$write and some_ed_data$prog
## Kruskal-Wallis chi-squared = 34.045, df = 2, p-value = 4.047e-08
```

McNemar test

```r
# Here is some made up data in matrix form
made_up_matrixdata <- matrix(c(150, 22, 21, 12), 2, 2)
mcnemar.test(made_up_matrixdata)
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  made_up_matrixdata
## McNemar's chi-squared = 0, df = 1, p-value = 1
```

Multiple Regression

```r
lm(some_ed_data$write ~ some_ed_data$female + some_ed_data$read +
     some_ed_data$math + some_ed_data$science + some_ed_data$socst)
```

```
##
## Call:
## lm(formula = some_ed_data$write ~ some_ed_data$female + some_ed_data$read +
##     some_ed_data$math + some_ed_data$science + some_ed_data$socst)
##
## Coefficients:
##          (Intercept)   some_ed_data$female     some_ed_data$read
##               6.1388                5.4925                0.1254
##    some_ed_data$math  some_ed_data$science    some_ed_data$socst
##               0.2381                0.2419                0.2293
```

Multiple Logistic Regression

```r
glm(some_ed_data$female ~ some_ed_data$read + some_ed_data$write,
    family = binomial)
```

```
##
## Call:  glm(formula = some_ed_data$female ~ some_ed_data$read + some_ed_data$write,
##     family = binomial)
##
## Coefficients:
##        (Intercept)  some_ed_data$read  some_ed_data$write
```

```
##          -1.70614             -0.07101              0.10637
##
## Degrees of Freedom: 199 Total (i.e. Null);  197 Residual
## Null Deviance:       275.6
## Residual Deviance: 247.8      AIC: 253.8
```

Multivariate Multiple Regression

```
mmrlm <- lm(cbind(write, read) ~ female + math + science + socst,
            data = some_ed_data)

summary(Anova(mmrlm))
```

```
##
## Type II MANOVA Tests:
##
## Sum of squares and products for error:
##          write     read
## write 7258.783 1091.057
## read  1091.057 8699.762
##
## ------------------------------------------
##
## Term: female
##
## Sum of squares and products for the hypothesis:
##           write       read
## write 1413.5284 -133.48461
## read  -133.4846   12.60544
##
## Multivariate Tests: female
##                  Df test stat approx F num Df den Df     Pr(>F)
## Pillai            1 0.1698853 19.85132      2    194 1.4335e-08 ***
## Wilks             1 0.8301147 19.85132      2    194 1.4335e-08 ***
## Hotelling-Lawley  1 0.2046528 19.85132      2    194 1.4335e-08 ***
## Roy               1 0.2046528 19.85132      2    194 1.4335e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------
##
## Term: math
##
## Sum of squares and products for the hypothesis:
##          write      read
## write 714.8665  856.2825
## read  856.2825 1025.6735
##
## Multivariate Tests: math
##                  Df test stat approx F num Df den Df     Pr(>F)
## Pillai            1 0.1599321 18.46685      2    194 4.5551e-08 ***
## Wilks             1 0.8400679 18.46685      2    194 4.5551e-08 ***
## Hotelling-Lawley  1 0.1903800 18.46685      2    194 4.5551e-08 ***
## Roy               1 0.1903800 18.46685      2    194 4.5551e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------
##
## Term: science
##
## Sum of squares and products for the hypothesis:
##          write     read
## write 857.8824 901.3191
## read  901.3191 946.9551
##
## Multivariate Tests: science
##                  Df test stat approx F num Df den Df     Pr(>F)
## Pillai            1 0.1664254 19.36631      2    194 2.1459e-08 ***
## Wilks             1 0.8335746 19.36631      2    194 2.1459e-08 ***
## Hotelling-Lawley  1 0.1996526 19.36631      2    194 2.1459e-08 ***
## Roy               1 0.1996526 19.36631      2    194 2.1459e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------
##
## Term: socst
##
## Sum of squares and products for the hypothesis:
##          write     read
## write 1105.653 1277.393
## read  1277.393 1475.810
##
## Multivariate Tests: socst
##                  Df test stat approx F num Df den Df     Pr(>F)
## Pillai            1 0.2206710 27.46604      2    194 3.1399e-11 ***
## Wilks             1 0.7793290 27.46604      2    194 3.1399e-11 ***
## Hotelling-Lawley  1 0.2831551 27.46604      2    194 3.1399e-11 ***
## Roy               1 0.2831551 27.46604      2    194 3.1399e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Non-parametric Correlation

```
cor.test(some_ed_data$read, some_ed_data$write, method = "spearman")
```

```
## Warning in cor.test.default(some_ed_data$read, some_ed_data$write, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  some_ed_data$read and some_ed_data$write
## S = 510993, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
```

```
## 0.6167455
```

One Sample *t*-test

```r
t.test(some_ed_data$read, mu = 50)
```

```
##
##  One Sample t-test
##
## data:  some_ed_data$read
## t = 3.0759, df = 199, p-value = 0.002394
## alternative hypothesis: true mean is not equal to 50
## 95 percent confidence interval:
##  50.80035 53.65965
## sample estimates:
## mean of x
##     52.23
```

One-way ANOVA (Analysis of Variance)

```r
summary(aov(some_ed_data$read ~ some_ed_data$prog))
```

```
##                    Df Sum Sq Mean Sq F value Pr(>F)
## some_ed_data$prog   1    381   381.1   3.674 0.0567 .
## Residuals         198  20538   103.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

One-way MANOVA (Multivariate Analysis of Variance)

```r
summary(manova(cbind(some_ed_data$read, some_ed_data$write, some_ed_data$math) ~
                 some_ed_data$prog))
```

```
##                    Df   Pillai approx F num Df den Df  Pr(>F)
## some_ed_data$prog   1 0.035319    2.392      3    196 0.06984 .
## Residuals         198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

One-way Repeated Measures ANOVA (Analysis of Variance)

```r
model <- lm(gender ~ item_1 + item_2, data = some_survey_data)

analysis <- Anova(model, idata = factor_surveydata, idesign = ~s)

print(analysis)
```

```
## Anova Table (Type II tests)
##
## Response: gender
##           Sum Sq Df F value Pr(>F)
## item_1    0.0601  1  0.2396 0.6307
## item_2    0.7268  1  2.8974 0.1069
```

```
## Residuals 4.2642 17
```

Ordered Logistic Regression

```
# Create ordered variable write_more as a factor with levels 1, 2, and 3
some_ed_data$write3 <- cut(some_ed_data$write, c(0, 48, 57, 70),  right = TRUE,
                           labels = c(1,2,3))

table(some_ed_data$write3)
```

```
##
##  1  2  3
## 61 61 78
```

```
# fit ordered logit model and store results 'some_write_data'
some_write_data <- polr(write3 ~ female + read + socst, data = some_ed_data,
                        Hess=TRUE)

summary(some_write_data)
```

```
## Call:
## polr(formula = write3 ~ female + read + socst, data = some_ed_data,
##     Hess = TRUE)
##
## Coefficients:
##           Value Std. Error t value
## female 1.28543    0.32445   3.962
## read   0.11772    0.02136   5.512
## socst  0.08019    0.01944   4.124
##
## Intercepts:
##     Value   Std. Error t value
## 1|2  9.7037  1.1968     8.1080
## 2|3 11.8001  1.3041     9.0486
##
## Residual Deviance: 312.5526
## AIC: 322.5526
```

Paired *t*-test

```
t.test(some_ed_data$write, some_ed_data$read, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  some_ed_data$write and some_ed_data$read
## t = 0.86731, df = 199, p-value = 0.3868
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6941424  1.7841424
## sample estimates:
## mean of the differences
##                   0.545
```

Principal Components Analysis

```r
princomp(formula = ~read + write + math + science + socst,
         data = some_ed_data)
```

```
## Call:
## princomp(formula = ~read + write + math + science + socst, data = some_ed_data)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5
## 18.252929  7.677044  6.213371  5.774331  5.429881
##
##  5  variables and  200 observations.
```

Repeated Measures Logistic Regression

```r
glmer(highpulse ~ diet + (1 | id), data = some_exercise_data,
      family = binomial)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: highpulse ~ diet + (1 | id)
##    Data: some_exercise_data
##      AIC      BIC   logLik deviance df.resid
## 105.4679 112.9674 -49.7340  99.4679       87
## Random effects:
##  Groups Name        Std.Dev.
##  id     (Intercept) 1.821
## Number of obs: 90, groups:  id, 30
## Fixed Effects:
## (Intercept)         diet
##      -3.148        1.145
```

Simple Linear Regression

```r
lm(some_ed_data$write ~ some_ed_data$read)
```

```
##
## Call:
## lm(formula = some_ed_data$write ~ some_ed_data$read)
##
## Coefficients:
##       (Intercept)  some_ed_data$read
##           23.9594             0.5517
```

Simple logistic regression

```r
glm(some_ed_data$female ~ some_ed_data$read, family = binomial)
```

```
##
## Call:  glm(formula = some_ed_data$female ~ some_ed_data$read, family = binomial)
##
```

```
## Coefficients:
##       (Intercept)  some_ed_data$read
##           0.72609           -0.01044
##
## Degrees of Freedom: 199 Total (i.e. Null);  198 Residual
## Null Deviance:       275.6
## Residual Deviance: 275.1      AIC: 279.1
```

Two independent samples *t*-test

```
t.test(some_ed_data$read ~ some_ed_data$female)
```

```
##
##  Welch Two Sample t-test
##
## data:  some_ed_data$read by some_ed_data$female
## t = 0.74506, df = 188.46, p-value = 0.4572
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -1.796263  3.976725
## sample estimates:
## mean in group 0 mean in group 1
##        52.82418        51.73394
```

Wilcoxon-Mann-Whitney Test

```
wilcox.test(some_ed_data$read ~ some_ed_data$female)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  some_ed_data$read by some_ed_data$female
## W = 5300, p-value = 0.4029
## alternative hypothesis: true location shift is not equal to 0
```

Wilcoxon Signed Rank Sum Test

```
wilcox.test(some_ed_data$write, some_ed_data$read, paired = TRUE)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  some_ed_data$write and some_ed_data$read
## V = 9261, p-value = 0.3666
## alternative hypothesis: true location shift is not equal to 0
```

A Final Rant About Open Source versus Proprietary

There are so many others like those dealing with Structural Equation Modeling (SEM) and a special case of this approach: Higher Linear Modeling (HLM), Machine Learning (ML) and Predictive Modeling (nope ML is NOT glorified statistics!), etc. Those softwares and add-ons become expensive as the methodology becomes specialized and companies/institutions/organizations are less likely to purchase them. If you want to be truly marketable and versatile, become proficient with an open-source software like R and Python. It will be worth it!

Side note

If you are a fan of the show Rick & Morty, consider downloading the most pointless package mortyr to do pointless statistics on pointless data. More about the package here.

If you're feeling confident, you can program an "AI" write its own pointless script of the show.