County Dashboard Comparison

```
## Loading required package: tidyverse

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'psych'
##
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
##
##
## Loading required package: carData
##
##
## Attaching package: 'car'
##
##
## The following object is masked from 'package:psych':
##
##     logit
##
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
##
## The following object is masked from 'package:purrr':
##
##     some
##
##
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
##
##
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
##
## Attaching package: 'rstatix'
##
##
## The following object is masked from 'package:stats':
##
##     filter
##
##
##
## Attaching package: 'effectsize'
##
##
## The following objects are masked from 'package:rstatix':
##
##     cohens_d, eta_squared
##
##
## The following object is masked from 'package:psych':
##
##     phi

## Warning: namespace 'ez' is not available and has been replaced
## by .GlobalEnv when processing object 'anova_result'
```

Load Data

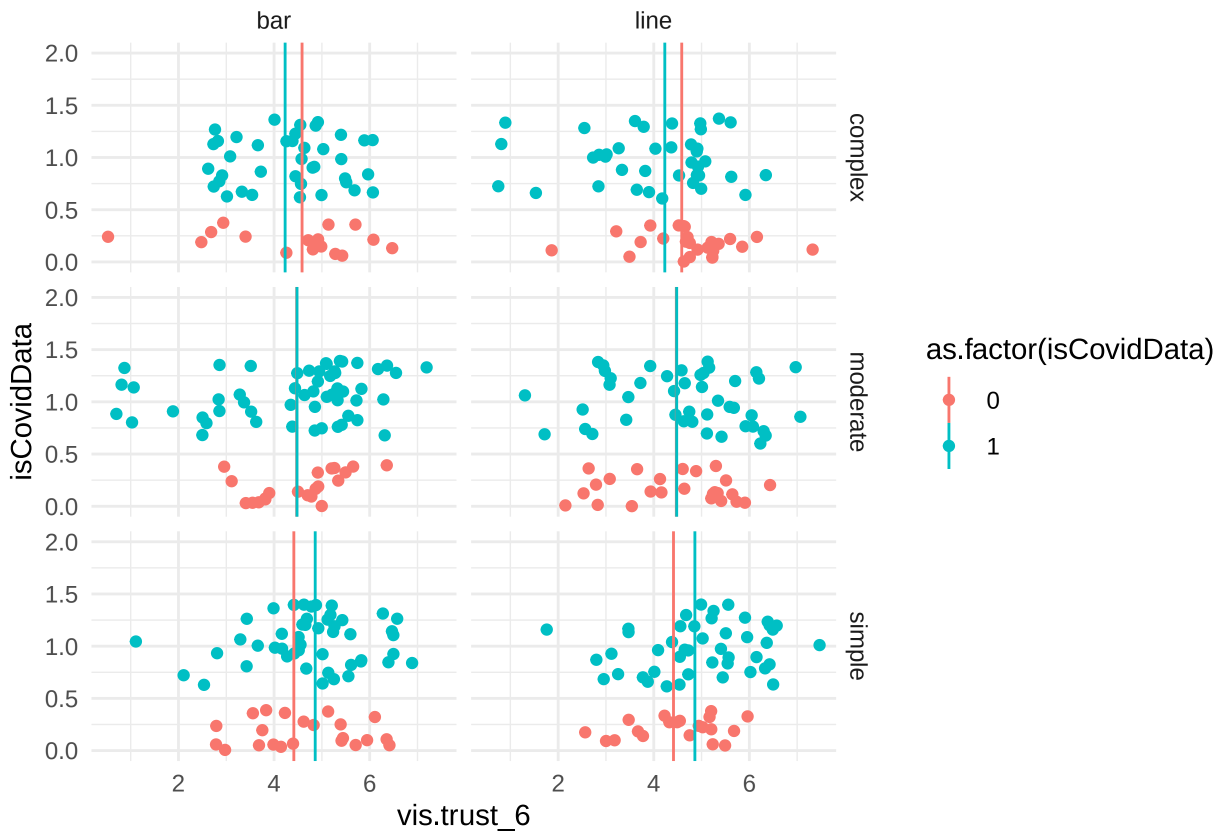
```
results <- read.csv("data_clean.csv")

# trust in data is the column: bar-data_6
# trust in vis is the column: bar-vis_6
```

Trust in Vis

```
results %>%
  ggplot(aes(x = vis.trust_6, y = isCovidData, colour = as.factor(isCovidData))) +
  geom_jitter(data = results, width = 0.5) +
  ylim(0, 2) +
  # labs(title = "Trust in data") +
  geom_vline(data = results %>%
               group_by(complexity, isCovidData) %>%
               summarize(n = n(),
                         vis.trust_6 = mean(vis.trust_6)),
             aes(xintercept = vis.trust_6, colour = as.factor(isCovidData))) +
  facet_grid(rows = vars(complexity), cols = vars(chartType)) +
  theme_minimal()
```

```
## `summarise()` has grouped output by 'complexity'. You can override using the
## `.groups` argument.
```

```
## Warning: Removed 144 rows containing missing values (‘geom_point()‘).
```




```
results %>%
  group_by(complexity, isCovidData) %>%
  summarize(n = n(),
            mean = mean(vis.trust_6),
            se = sd(vis.trust_6)/sqrt(n),
            n = n)
```

```
## ‘summarise()‘ has grouped output by ’complexity’. You can override using the
## ‘.groups‘ argument.
```

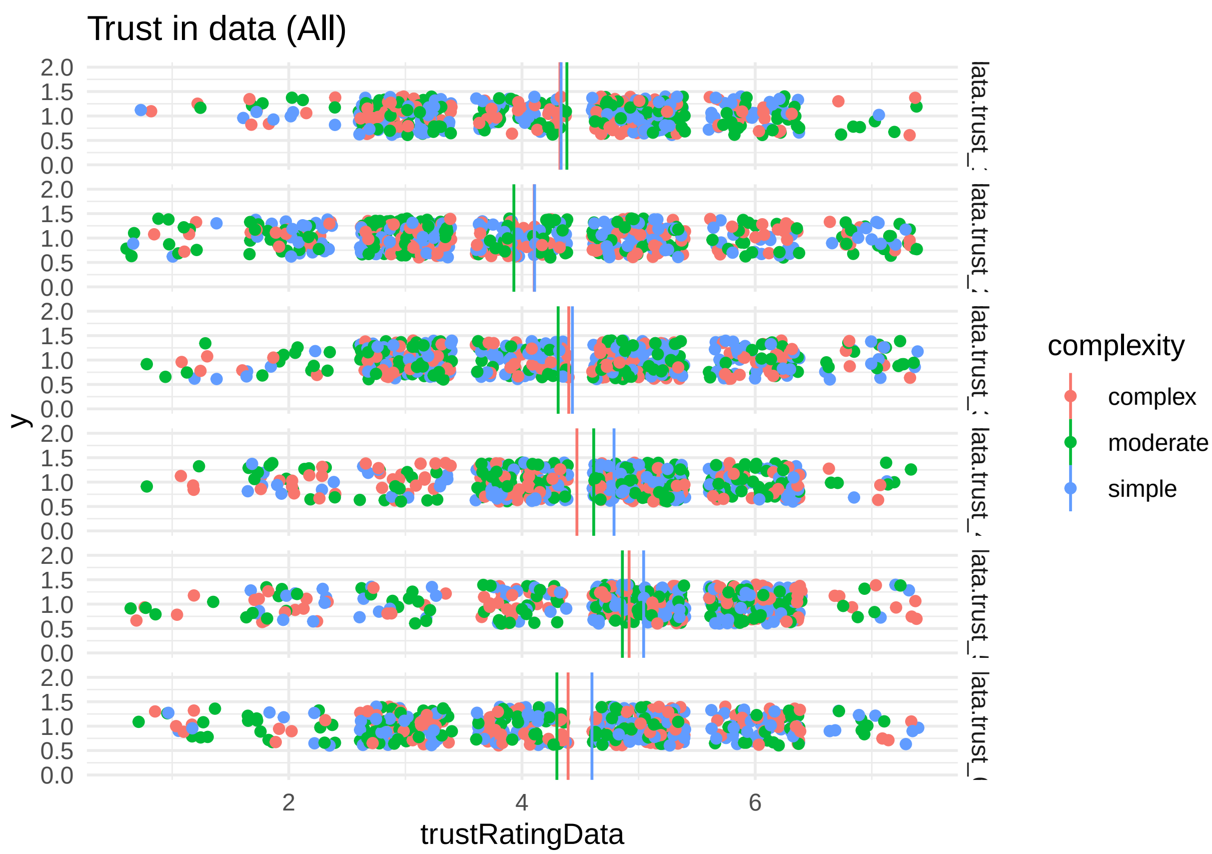
```
## # A tibble: 6 x 5
## # Groups:   complexity [3]
##   complexity isCovidData     n  mean    se
##   <chr>            <int> <int> <dbl> <dbl>
## 1 complex              0    94  4.59 0.122
## 2 complex              1    78  4.23 0.138
## 3 moderate             0    89  4.47 0.114
## 4 moderate             1    98  4.48 0.149
## 5 simple               0    92  4.41 0.106
## 6 simple               1    93  4.86 0.124
```

```r
model <- lm(formula = vis.trust_6 ~ complexity * as.factor(isCovidData) + chartType
                              + Age + Gender + State_1 + Education + Parents_education + Language
            data = results)
anova(model)
```

```
## Analysis of Variance Table
##
## Response: vis.trust_6
##                                    Df Sum Sq Mean Sq F value   Pr(>F)
## complexity                          2   4.49  2.2440  1.5764 0.207781
## as.factor(isCovidData)              1   0.26  0.2599  0.1826 0.669362
## chartType                           1   0.56  0.5579  0.3919 0.531598
## Age                                 1   2.78  2.7802  1.9530 0.162904
## Gender                              1   2.89  2.8936  2.0327 0.154595
## State_1                            45  84.39  1.8752  1.3173 0.087260 .
## Education                           1   0.04  0.0363  0.0255 0.873171
## Parents_education                   1   2.97  2.9685  2.0853 0.149369
## Language                            1   0.14  0.1417  0.0995 0.752535
## Ethnicity                           1   2.71  2.7093  1.9032 0.168354
## Income                              1   0.10  0.1019  0.0715 0.789209
## Religion                            1   0.29  0.2900  0.2037 0.651948
## complexity:as.factor(isCovidData)   2  13.28  6.6401  4.6645 0.009852 **
## Residuals                         484 688.99  1.4235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
results_long_data <- results %>%
  select(data.trust_1, data.trust_2, data.trust_3,
         data.trust_4, data.trust_5, data.trust_6,
         ResponseId, complexity,
         vlat_simple, vlat_moderate, vlat_complex) %>%
  gather(key = trustItemData, value = trustRatingData,
         data.trust_1, data.trust_2, data.trust_3, data.trust_4, data.trust_5, data.trust_6)

results_long_data %>%
  ggplot(aes(x = trustRatingData, y = 1, color = complexity)) +
  geom_jitter() +
  ylim(0, 2) +
  labs(title = "Trust in data (All)") +
  geom_vline(data = results_long_data %>%
               group_by(complexity, trustItemData) %>%
               summarize(n = n(),
                         average = mean(trustRatingData)),
             aes(xintercept = average, color = complexity)) +
  facet_grid(vars(trustItemData)) +
  theme_minimal()
```

```
## `summarise()` has grouped output by 'complexity'. You can override using the
## `.groups` argument.
```



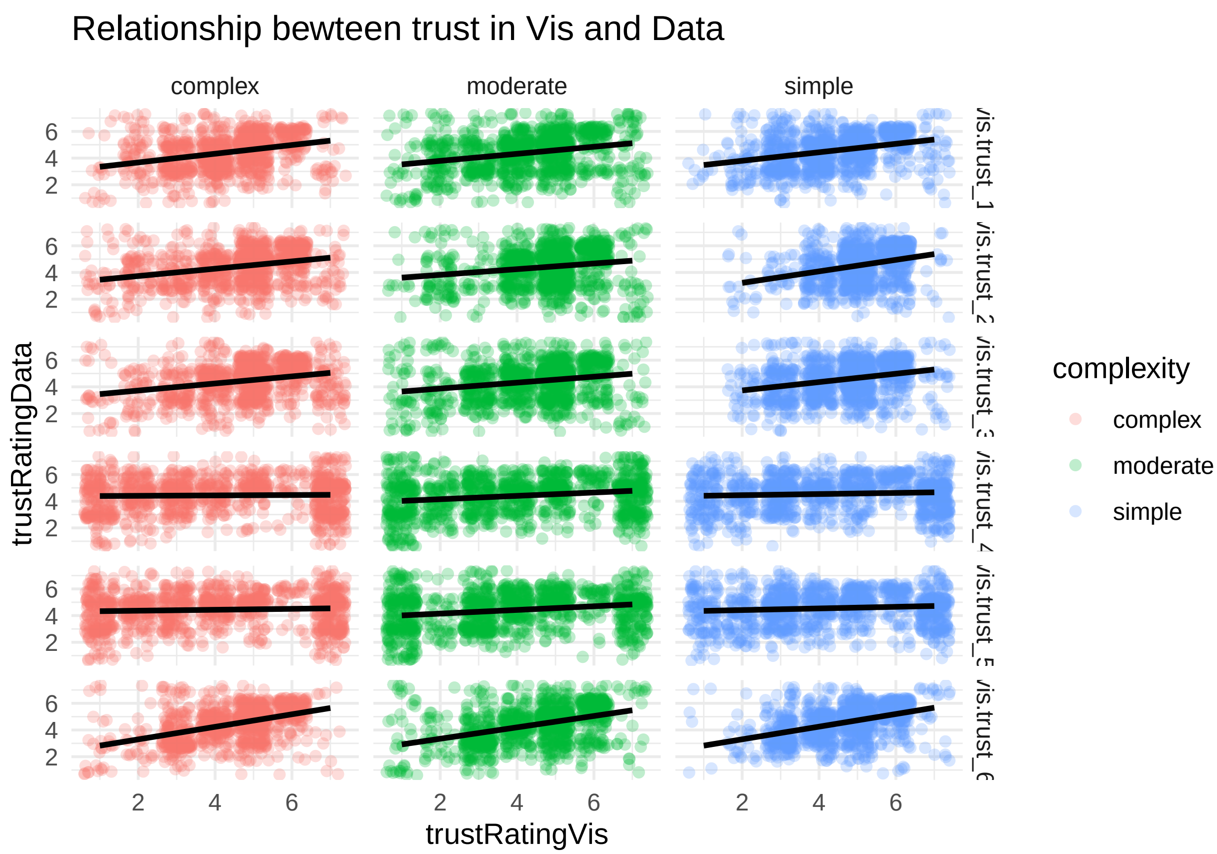
Relationship between trust in data and vis, across complexity

```
results_long_data <- results %>%
  select(data.trust_1, data.trust_2, data.trust_3,
         data.trust_4, data.trust_5, data.trust_6,
         ResponseId, complexity,
         vlat_simple, vlat_moderate, vlat_complex) %>%
  gather(key = trustItemData, value = trustRatingData,
         data.trust_1, data.trust_2, data.trust_3, data.trust_4, data.trust_5, data.trust_6)

results_long_vis <- results %>%
  gather(key = trustItemVis, value = trustRatingVis,
         vis.trust_1, vis.trust_2, vis.trust_3, vis.trust_4, vis.trust_5, vis.trust_6)

results_long_all <- merge(results_long_vis, results_long_data,
                          by = c("ResponseId", "complexity",
                                 "vlat_simple", "vlat_moderate", "vlat_complex"))

model<- glm(trustRatingData ~  trustRatingVis * complexity + chartType,
                      data = results_long_all)
summary(model)
```

```
##
```

```
## Call:
## glm(formula = trustRatingData ~ trustRatingVis * complexity +
##     chartType, data = results_long_all)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9514  -0.9377   0.3009   0.8100   3.2526
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        3.781523   0.045801  82.563  < 2e-16 ***
## trustRatingVis                     0.143167   0.009675  14.798  < 2e-16 ***
## complexitymoderate                -0.224473   0.062616  -3.585 0.000338 ***
## complexitysimple                  -0.006971   0.067263  -0.104 0.917460
## chartTypeline                      0.061947   0.018391   3.368 0.000758 ***
## trustRatingVis:complexitymoderate  0.047174   0.013535   3.485 0.000493 ***
## trustRatingVis:complexitysimple    0.021261   0.014215   1.496 0.134755
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.6436)
##
##     Null deviance: 33729  on 19583  degrees of freedom
## Residual deviance: 32177  on 19577  degrees of freedom
## AIC: 65317
##
## Number of Fisher Scoring iterations: 2
```

```
results_long_all %>%
  # filter(trustItemVis == "bar.vis_2") %>%
  ggplot(aes(x = trustRatingVis, y = trustRatingData, color = complexity)) +
  geom_jitter(alpha = 0.25) +
  stat_smooth(method = "lm",
              formula = y ~ x,
              geom = "smooth", color = "black") +
  labs(title = "Relationship bewteen trust in Vis and Data") +
  facet_grid(rows = vars(trustItemVis), cols = vars(complexity)) +
  theme_minimal()
```



Does the trust items predict trust?

```
model <- lm(formula = vis.trust_6 ~ vis.trust_1 + vis.trust_2 + vis.trust_3 + vis.trust_4 + vis.trust_5
            data = results)
anova(model)
```

```
## Analysis of Variance Table
##
## Response: vis.trust_6
##              Df Sum Sq Mean Sq  F value    Pr(>F)
## vis.trust_1   1 152.28 152.282 152.3443 < 2.2e-16 ***
## vis.trust_2   1  50.46  50.457  50.4773 3.842e-12 ***
## vis.trust_3   1  60.61  60.607  60.6314 3.558e-14 ***
## vis.trust_4   1   2.42   2.419   2.4201    0.1204
## vis.trust_5   1   0.34   0.338   0.3382    0.5611
## Residuals   538 537.78   1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model <- lm(formula = data.trust_6 ~ data.trust_1 + data.trust_2 + data.trust_3 + data.trust_4 + data.t
            data = results)
anova(model)
```

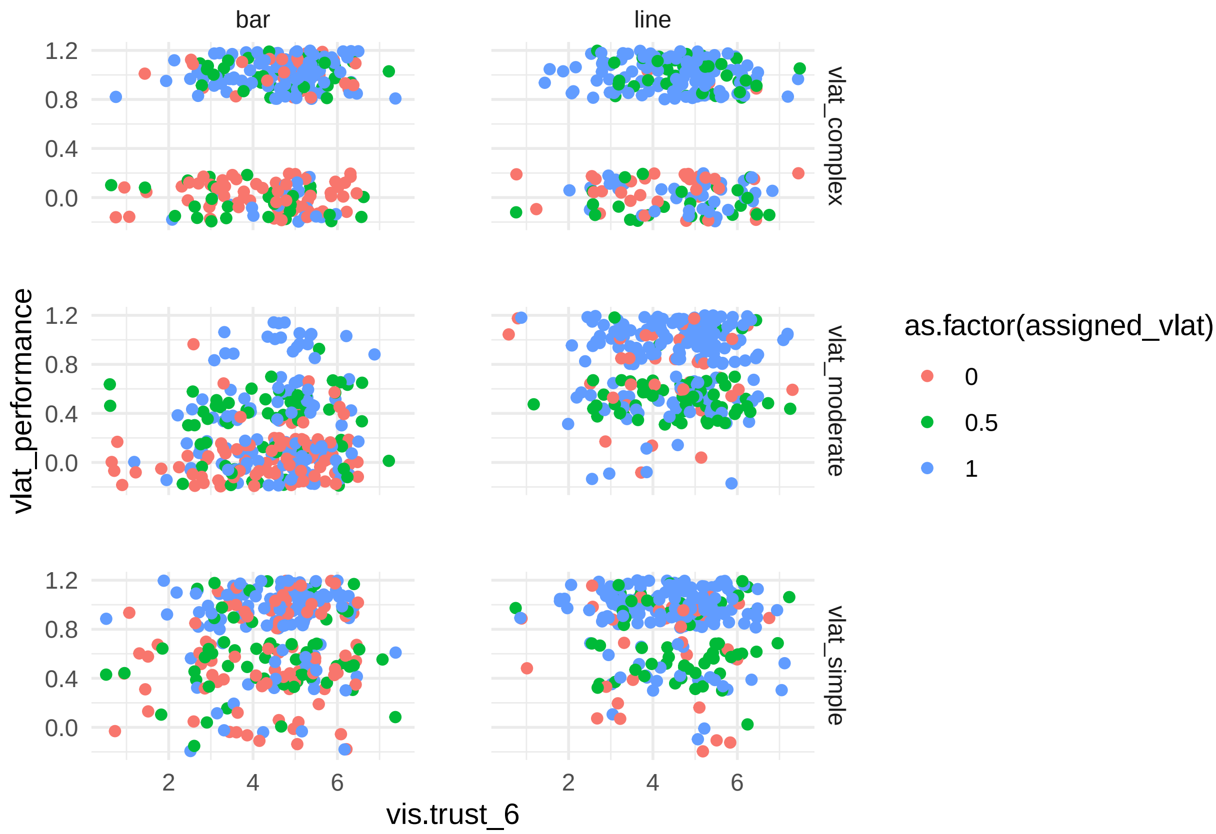
```
## Analysis of Variance Table
##
## Response: data.trust_6
##                Df Sum Sq Mean Sq  F value    Pr(>F)
## data.trust_1    1 202.15 202.146 179.7603 < 2.2e-16 ***
## data.trust_2    1  17.19  17.186  15.2829 0.0001044 ***
## data.trust_3    1  63.38  63.379  56.3602 2.521e-13 ***
## data.trust_4    1  15.80  15.796  14.0467 0.0001976 ***
## data.trust_5    1   1.98   1.980   1.7605 0.1851220
## Residuals     538 605.00   1.125
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How does performance on VLAT questions predict trust?

```r
model <- lm(formula = vis.trust_6 ~ vlat_simple + vlat_moderate + vlat_complex,
            data = results)
anova(model)
```

```
## Analysis of Variance Table
##
## Response: vis.trust_6
##                Df Sum Sq Mean Sq F value Pr(>F)
## vlat_simple     1   1.64  1.6368  1.1078 0.2930
## vlat_moderate   1   0.93  0.9338  0.6320 0.4270
## vlat_complex    1   3.49  3.4893  2.3617 0.1249
## Residuals     540 797.82  1.4774
```

```r
results %>%
  gather(key = vlat_level, value = vlat_performance, vlat_simple,  vlat_moderate, vlat_complex) %>%
  ggplot(aes(x = vis.trust_6, y = vlat_performance, colour = as.factor(assigned_vlat))) +
  geom_jitter(width = 0.5) +
  # y(0, 2) +
  # labs(title = "Trust in data") +
  # geom_vline(data = results %>%
  #              group_by(complexity, isCovidData) %>%
  #              summarize(n = n(),
  #                        vis.trust_6 = mean(vis.trust_6)),
  #            aes(xintercept = vis.trust_6, colour = as.factor(isCovidData))) +
  facet_grid(rows = vars(vlat_level), cols = vars(chartType)) +
  theme_minimal() +
  theme(panel.spacing = unit(2, "lines"))
```




```
        # legend.position = "none")
```

How does provenance data predict trust?

Hypothesis: for the complex condition, we expect people who brushed more to have higher trust

```
complexCondition <- results %>%
  filter(complexity == "complex")

# can change the predictor to bar.vis
model<- manova(cbind(data.trust_1,
                     data.trust_2,
                     data.trust_3,
                     data.trust_4,
                     data.trust_5,
                     data.trust_6) ~ brushed + explore_interactions + explore_time,
               data = complexCondition)
summary.aov(model)
```

```
##  Response data.trust_1 :
##                          Df  Sum Sq Mean Sq F value Pr(>F)
## brushed                   1   0.014 0.01358  0.0092 0.9236
## explore_interactions      1   0.005 0.00505  0.0034 0.9534
```

```
## explore_time           1   0.260 0.26008  0.1765 0.6749
## Residuals            168 247.489 1.47315
##
##  Response data.trust_2 :
##                        Df Sum Sq Mean Sq F value Pr(>F)
## brushed                 1   0.44 0.44061  0.2290 0.6329
## explore_interactions   1   1.70 1.70271  0.8850 0.3482
## explore_time           1   0.73 0.73208  0.3805 0.5382
## Residuals            168 323.24 1.92405
##
##  Response data.trust_3 :
##                        Df  Sum Sq Mean Sq F value Pr(>F)
## brushed                 1   1.331 1.33131  0.7957 0.3737
## explore_interactions   1   0.712 0.71243  0.4258 0.5149
## explore_time           1   0.186 0.18588  0.1111 0.7393
## Residuals            168 281.090 1.67316
##
##  Response data.trust_4 :
##                        Df  Sum Sq Mean Sq F value Pr(>F)
## brushed                 1   0.436 0.43648  0.2907 0.5905
## explore_interactions   1   0.143 0.14305  0.0953 0.7580
## explore_time           1   0.043 0.04309  0.0287 0.8657
## Residuals            168 252.232 1.50138
##
##  Response data.trust_5 :
##                        Df  Sum Sq Mean Sq F value Pr(>F)
## brushed                 1   2.990 2.99005  1.7740 0.1847
## explore_interactions   1   0.402 0.40205  0.2385 0.6259
## explore_time           1   2.304 2.30354  1.3667 0.2440
## Residuals            168 283.165 1.68550
##
##  Response data.trust_6 :
##                        Df  Sum Sq Mean Sq F value Pr(>F)
## brushed                 1   1.235 1.23469  0.8146 0.3680
## explore_interactions   1   0.078 0.07843  0.0517 0.8203
## explore_time           1   1.175 1.17467  0.7750 0.3799
## Residuals            168 254.628 1.51565
```

Hypothesis: for all the conditions, we expect people who hovered more to have higher trust

```
# can change the predictor to bar.vis
model<- manova(cbind(vis.trust_6,
                     vis.trust_5,
                     vis.trust_4,
                     vis.trust_3,
                     vis.trust_2,
                     vis.trust_1) ~ brushed + explore_interactions + explore_time,
               data = results)
summary.aov(model)
```

```
##  Response vis.trust_6 :
##                        Df Sum Sq Mean Sq F value Pr(>F)
## brushed                 1   0.22 0.22139  0.1488 0.6999
```

```
## explore_interactions   1   0.02 0.02019  0.0136 0.9073
## explore_time           1   0.02 0.01509  0.0101 0.9198
## Residuals            540 803.63 1.48820
##
##  Response vis.trust_5 :
##                        Df  Sum Sq Mean Sq F value Pr(>F)
## brushed                1    0.31  0.3144  0.0725 0.7878
## explore_interactions   1    6.40  6.3983  1.4761 0.2249
## explore_time           1   10.47 10.4729  2.4160 0.1207
## Residuals            540 2340.75  4.3347
##
##  Response vis.trust_4 :
##                        Df  Sum Sq Mean Sq F value  Pr(>F)
## brushed                1    1.91  1.9067  0.4068 0.52388
## explore_interactions   1    0.67  0.6694  0.1428 0.70564
## explore_time           1   13.54 13.5401  2.8887 0.08978 .
## Residuals            540 2531.09  4.6872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response vis.trust_3 :
##                        Df Sum Sq Mean Sq F value  Pr(>F)
## brushed                1   5.18  5.1797  3.0917 0.07926 .
## explore_interactions   1   1.96  1.9620  1.1711 0.27965
## explore_time           1   4.17  4.1744  2.4917 0.11504
## Residuals            540 904.68  1.6753
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response vis.trust_2 :
##                        Df Sum Sq Mean Sq F value  Pr(>F)
## brushed                1   4.90  4.9035  3.4754 0.06283 .
## explore_interactions   1   0.05  0.0518  0.0367 0.84818
## explore_time           1   1.50  1.5044  1.0663 0.30225
## Residuals            540 761.89  1.4109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response vis.trust_1 :
##                        Df Sum Sq Mean Sq F value Pr(>F)
## brushed                1   0.21 0.20567  0.1198 0.7294
## explore_interactions   1   0.08 0.08061  0.0470 0.8285
## explore_time           1   0.05 0.05321  0.0310 0.8603
## Residuals            540 927.10 1.71685
```

How does trust in science predict trust?

How does need for cognition predict trust?

Power Analysis

```r
eta_squared(model, partial = TRUE)
```

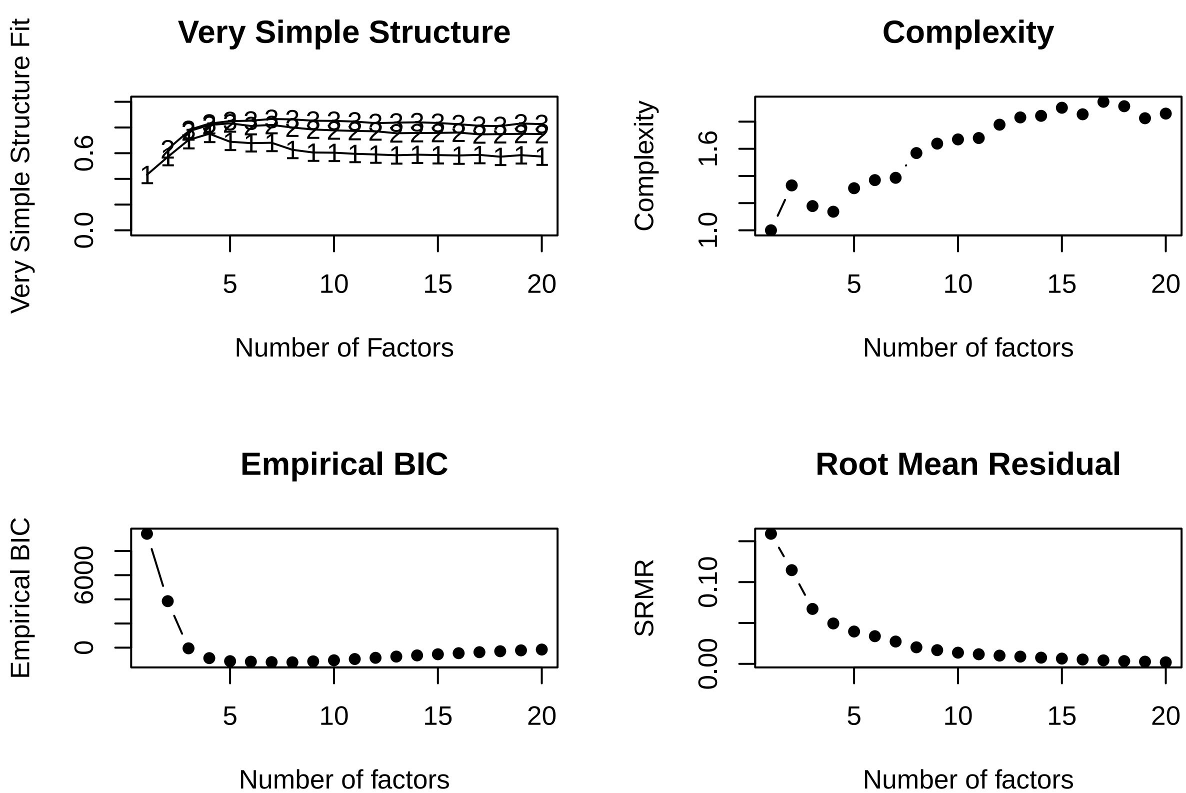
```
## # Effect Size for ANOVA (Type I)
##
## Parameter              | Eta2 (partial) |       95% CI
## -------------------------------------------------------
## brushed                |           0.02 | [0.00, 1.00]
## explore_interactions   |       6.59e-03 | [0.00, 1.00]
## explore_time           |           0.02 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].
```

Factor Analysis

```r
factorAnalysis <- results %>%
  select(data.trust_1, data.trust_2, data.trust_3, data.trust_4, data.trust_5, data.trust_6,
         vis.trust_1, vis.trust_2, vis.trust_3, vis.trust_4, vis.trust_5, vis.trust_6,
         trust.in.science_1, trust.in.science_2, trust.in.science_3, trust.in.science_4, trust.in.scienc
         trust.in.science_6, trust.in.science_7, trust.in.science_8,
         cognition_1, cognition_2, cognition_3, cognition_4, cognition_5, cognition_6,
         brushed, explore_interactions, explore_time,
         interpersonal.trust_1)

nfactors(factorAnalysis)
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully
```




```
##
## Number of factors
## Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
##     n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
```

```
## VSS complexity 1 achieves a maximimum of 0.75  with  4  factors
## VSS complexity 2 achieves a maximimum of 0.83  with  5  factors
## The Velicer MAP achieves a minimum of 0.01  with  3  factors
## Empirical BIC achieves a minimum of  -1210.42  with  8  factors
## Sample Size adjusted BIC achieves a minimum of  -331.97  with  9  factors
##
## Statistics by number of factors
##    vss1 vss2   map dof chisq     prob sqresid  fit  RMSEA  BIC SABIC complex
## 1  0.43 0.00 0.043 405  5598  0.0e+00    43.1 0.43 0.1535 3047  4332     1.0
## 2  0.57 0.63 0.034 376  3664  0.0e+00    28.0 0.63 0.1268 1295  2489     1.3
## 3  0.70 0.77 0.015 348  2182 1.7e-262    16.7 0.78 0.0984  -10  1095     1.2
## 4  0.75 0.82 0.015 321  1433 5.9e-140    12.8 0.83 0.0798 -589   430     1.1
## 5  0.69 0.83 0.016 295  1094  3.9e-92    10.9 0.86 0.0705 -764   172     1.3
## 6  0.68 0.81 0.017 270   740  2.0e-45     9.8 0.87 0.0566 -960  -103     1.4
## 7  0.68 0.82 0.018 246   603  6.2e-32     8.4 0.89 0.0516 -947  -166     1.4
## 8  0.63 0.80 0.018 223   418  5.5e-14     7.4 0.90 0.0400 -987  -279     1.6
## 9  0.61 0.78 0.021 201   296  1.4e-05     6.9 0.91 0.0294 -970  -332     1.6
## 10 0.60 0.78 0.023 180   248  6.0e-04     6.2 0.92 0.0263 -886  -315     1.7
## 11 0.60 0.77 0.026 160   213  3.2e-03     5.6 0.93 0.0246 -795  -287     1.7
## 12 0.59 0.77 0.030 141   176  2.3e-02     5.1 0.93 0.0214 -712  -264     1.8
## 13 0.58 0.76 0.037 123   149  5.8e-02     4.9 0.93 0.0195 -626  -236     1.8
## 14 0.59 0.76 0.043 106   107  4.5e-01     4.7 0.94 0.0041 -561  -224     1.8
## 15 0.59 0.76 0.052  90    82  7.1e-01     4.6 0.94 0.0000 -485  -199     1.9
## 16 0.58 0.76 0.061  75    66  7.6e-01     4.3 0.94 0.0000 -406  -168     1.9
## 17 0.59 0.75 0.073  61    45  9.4e-01     4.2 0.94 0.0000 -339  -146     1.9
## 18 0.57 0.75 0.073  48    29  9.9e-01     4.0 0.95 0.0000 -274  -121     1.9
## 19 0.58 0.75 0.085  36    19  9.9e-01     4.1 0.95 0.0000 -207   -93     1.8
## 20 0.57 0.75 0.094  25    12  9.9e-01     3.7 0.95 0.0000 -146   -66     1.9
##     eChisq   SRMR  eCRMS  eBIC
## 1  11977.6 0.1591 0.1649  9427
## 2   6216.9 0.1146 0.1233  3848
## 3   2136.4 0.0672 0.0751   -56
## 4   1155.8 0.0494 0.0575  -866
## 5    740.7 0.0396 0.0480 -1118
## 6    541.7 0.0338 0.0429 -1159
## 7    353.0 0.0273 0.0363 -1197
## 8    194.2 0.0203 0.0283 -1210
## 9    133.1 0.0168 0.0247 -1133
## 10    88.9 0.0137 0.0213 -1045
## 11    65.4 0.0118 0.0194  -942
## 12    48.4 0.0101 0.0178  -840
## 13    37.3 0.0089 0.0167  -737
## 14    27.3 0.0076 0.0154  -640
## 15    19.7 0.0064 0.0142  -547
## 16    13.5 0.0053 0.0128  -459
## 17     8.8 0.0043 0.0115  -375
## 18     5.4 0.0034 0.0102  -297
## 19     3.5 0.0027 0.0094  -223
## 20     1.8 0.0019 0.0081  -156
```

```
f5 <- fa(factorAnalysis, 5)
```

```
## Loading required namespace: GPArotation
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect. Try a
## different factor score estimation method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected. Examine the results carefully
```

```
pdf(file = "factorAnalysisDiagramF3.pdf",   # The directory you want to save the file in
    width = 15, # The width of the plot in inches
    height = 23) # The height of the plot in inches
fa.diagram(f5)
dev.off()
```

```
## pdf
##   2
```

```
# based on the factor analysis, it looks like not all the vis Qs go together and not all the data Qs go
```