SOC 325: Quantified-Self

Basics of Statistics

Zack W Almquist
University of Washington

2023-01-18

Road Map

Section Introduction

R Lab

- No R Lab this week
- Everyone gets full credit
- Extra week with Lab 1

Overview

- In this lecture we are going to review basic statistics with R
  - Central Tendencies (e.g. mean, median, mode)
  - Volatility (e.g. variance, sd)
  - Statistical test (e.g. permutation and t-test)
  - Linear Regression

Data

Data

Half Marathon Times at Cherry Blossom Run

Data: Overview

- Results for Seattle Cherry Blossom Run 2022
  - I used Chrome to download an archive of these results
  - To directly scrape is a bit complicated because you have to get the javascript to precompile
- We will use the following packages to read and manipulate the data

```
library(tidyverse)
library(textreadr)
library(rvest)
library(lubridate)
library(AMR)
library(here)
```

Data: Import and clean

```r
## Read in the archived results
cb<-rvest::read_html(here("data/","half", "Seattle Cherry Blossom Run Results.html"))
## Clean and pull the table
cb_tab<-cb%>% minimal_html()%>%
  html_node("table") %>%
  html_table(header=1)

## Clean up name and add time in minutes
cb_tab<-cb_tab%>%mutate(
  ChipTime_min=hms(ChipTime),
  ChipTime_min=hour(ChipTime_min)*60 + minute(ChipTime_min)+second(ChipTime_min)/100,
  Name=substr(str_remove_all(str_remove_all(Name,"\\n"),"\\t"),2,1000000L)
)

## Add age groups in 5s
cb_tab$age_groups<-AMR::age_groups(cb_tab$Age,"fives")
```

Data: Import and clean

```
glimpse(cb_tab)
```

```
## Rows: 1,198
## Columns: 13
## $ Place             <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ Bib               <int> 19025, 17022, 17561, 17591, 17659, 17614, 17133, 171~
## $ Name              <chr> "KyleCochrun", "JuliaBudniak", "HaroldDanks", "Jared~
## $ Gender            <chr> "M", "F", "M", "M", "M", "M", "M", "M", "M", "M", "M~
## $ City              <chr> "Seattle", "Lakewood", "Chattanooga", "Washington", ~
## $ State             <chr> "WA", "WA", "TN", "DC", "WA", "WA", "WA", "WA", "WA"~
## $ ChipTime          <chr> "1:19:46.38", "1:21:18.23", "1:21:39.60", "1:21:45.9~
## $ Age               <int> 29, 41, 25, 41, 12, 24, 36, 26, 27, 26, 29, 32, 33, ~
## $ DivisionPlace     <int> 1, 1, 2, 1, 1, 1, 1, 3, 5, 4, 6, 1, 2, 3, 8, 7, 2, 2~
## $ Division          <chr> "M 25-29", "F 40-44", "M 25-29", "M 40-44", "M 10-14~
## $ `Group/Team Name` <chr> "", "", "", "", "", "", "Seattle Frontrunners", "Sea~
## $ ChipTime_min      <dbl> 79.4638, 81.1823, 81.3960, 81.4590, 82.3811, 83.3603~
## $ age_groups        <ord> 25-29, 40-44, 25-29, 40-44, 10-14, 20-24, 35-39, 25-~
```

Descriptive Statistics

Descriptive Statistics

```
## Counts by age
cb_tab%>%group_by(age_groups)%>%summarise(n=n())
```

```
## # A tibble: 13 x 2
##    age_groups     n
##    <ord>      <int>
##  1 10-14          5
##  2 15-19         19
##  3 20-24        127
##  4 25-29        275
##  5 30-34        214
##  6 35-39        166
##  7 40-44        143
##  8 45-49         89
##  9 50-54         80
## 10 55-59         37
## 11 60-64         28
## 12 65-69         12
## 13 70-74          3
```

Descriptive Statistics

```r
## Counts by age
cb_tab%>%group_by(age_groups,Gender)%>%summarise(n=n())
```

```
## # A tibble: 32 x 3
## # Groups:   age_groups [13]
##    age_groups Gender     n
##    <ord>      <chr>  <int>
##  1 10-14      "F"        3
##  2 10-14      "M"        2
##  3 15-19      "F"        9
##  4 15-19      "M"       10
##  5 20-24      ""         1
##  6 20-24      "F"       63
##  7 20-24      "M"       63
##  8 25-29      ""         3
##  9 25-29      "F"      122
## 10 25-29      "M"      148
## # ... with 22 more rows
```

Descriptive Statistics

```
## Counts by age
cb_tab%>%group_by(age_groups,Gender)%>%summarise(n=n())
```

```
## # A tibble: 32 x 3
## # Groups:   age_groups [13]
##    age_groups Gender     n
##    <ord>      <chr>  <int>
##  1 10-14      "F"        3
##  2 10-14      "M"        2
##  3 15-19      "F"        9
##  4 15-19      "M"       10
##  5 20-24      ""         1
##  6 20-24      "F"       63
##  7 20-24      "M"       63
##  8 25-29      ""         3
##  9 25-29      "F"      122
## 10 25-29      "M"      148
## # ... with 22 more rows
```

Descriptive Statistics

Update data: Limit to just “M” and “F” cases

```
cb_tab_mf<-cb_tab%>%filter(Gender%in%c("F","M"))
```

Descriptive Statistics

```r
## Age groups and Gender
cb_tab_mf%>%group_by(age_groups,Gender)%>%summarise(n=n())
```

```
## # A tibble: 26 x 3
## # Groups:   age_groups [13]
##    age_groups Gender     n
##    <ord>      <chr>  <int>
##  1 10-14      F          3
##  2 10-14      M          2
##  3 15-19      F          9
##  4 15-19      M         10
##  5 20-24      F         63
##  6 20-24      M         63
##  7 25-29      F        122
##  8 25-29      M        148
##  9 30-34      F        106
## 10 30-34      M        106
## # ... with 16 more rows
```

Descriptive Statistics

```
stats_overall<-cb_tab_mf%>%summarise(
  Total=n(),
  AvgTime=mean(ChipTime_min),
  se=sd(ChipTime_min)/Total,
  med=median(ChipTime_min),
  q5=quantile(ChipTime_min,.05),
  q95=quantile(ChipTime_min,.95)
)

## Print
stats_overall
```

```
## # A tibble: 1 x 6
##   Total AvgTime     se   med    q5   q95
##   <int>   <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1  1188    138. 0.0280  131.  96.3  203.
```

Descriptive Statistics

```
stats_gender<-cb_tab_mf%>%group_by(Gender)%>%summarise(
  Total=n(),
  AvgTime=mean(ChipTime_min),
  se=sd(ChipTime_min)/Total,
  med=median(ChipTime_min),
  q5=quantile(ChipTime_min,.05),
  q95=quantile(ChipTime_min,.95)
)

## Print
stats_gender
```

```
## # A tibble: 2 x 7
##   Gender Total AvgTime     se   med    q5   q95
##   <chr>  <int>   <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 F        637    146. 0.0532  137. 109.   216.
## 2 M        551    128. 0.0537  120.  89.8  184.
```

Statistical Inference

Statistical Inference

We will use the `infer` package

```
library(infer)
```

Statistical Inference

- Mean: $\bar{x} = \frac{1}{n}\sum_1^n x_i$
- We will use a simple permutation/simulation test

```
# calculate the observed statistic
observed_statistic <- cb_tab_mf %>%
  specify(response = ChipTime_min) %>%
  calculate(stat = "mean")

observed_statistic
```

```
## Response: ChipTime_min (numeric)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  138.
```

Statistical Inference

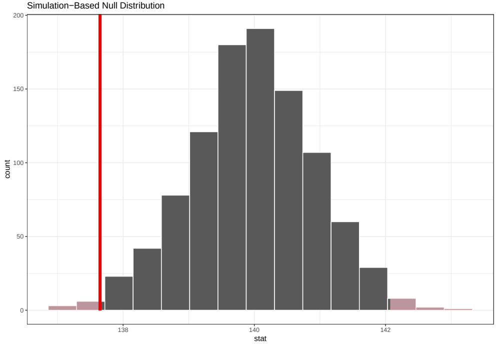
- Mean: $\bar{x} = \frac{1}{n}\sum_1^n x_i$
- The average Female Time for a half Marathon for 20-24 year olds is 140 Min
  - Our null hypthesis is that average time for a runner in the CBR is 140 Min
- We will generate a plausable null distribution from this assumption

```
# generate the null distribution
null_dist_1_sample <- cb_tab_mf %>%
  specify(response = ChipTime_min) %>%
  hypothesize(null = "point", mu = 140) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "mean")
```

Statistical Inference

- Mean: $\bar{x} = \frac{1}{n}\sum_1^n x_i$

```r
# visualize the null distribution and test statistic!
null_dist_1_sample %>%
  visualize() +
  shade_p_value(observed_statistic,
                direction = "two-sided")+
  theme_bw()
```



Statistical Inference

- Mean: $\bar{x} = \frac{1}{n}\sum_1^n x_i$

```
# calculate the p value from the test statistic and null distribution
p_value_1_sample <- null_dist_1_sample %>%
  get_p_value(obs_stat = observed_statistic,
              direction = "two-sided")

p_value_1_sample
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.014
```

Statistical Inference

- Mean: $\bar{x} = \frac{1}{n}\sum_1^n x_i$
- Instead of the permutation/simulation test we can do a `t-test`
  - Approximate normal test
- $t = \frac{\bar{x}}{se(\bar{x})}$
  - $se(\bar{x}) = sd(x)/\sqrt{n}$
  - $sd(x) = \sqrt{\frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})^2}$ (mle)
  - $sd(x) = \sqrt{\frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{x})^2}$ (unbiased)

```
##t.test
cb_tab_mf%>%t_test(response =ChipTime_min , mu = 140)
```

```
## # A tibble: 1 x 7
##   statistic  t_df p_value alternative estimate lower_ci upper_ci
##       <dbl> <dbl>   <dbl> <chr>          <dbl>    <dbl>    <dbl>
## 1     -2.44  1187  0.0149 two.sided       138.     136.     140.
```

Statistical Inference

- P-value of the t-statistic
  - Provides simple decision heuristic for `t-test`
  - p-value < 0.05 - statistically significant (there is a difference!)
  - p-value > 0.05 - statistically not significant (there is **not** a difference!)

```
# calculate the observed statistic
observed_statistic <- cb_tab_mf %>%
  specify(response = ChipTime_min) %>%
  hypothesize(null = "point", mu = 40) %>%
  calculate(stat = "t") %>%
  dplyr::pull()

# calculate 2-tail t-test
pt(observed_statistic, df = nrow(gss) - 1, lower.tail = FALSE)*2
```

```
## t
## 0
```

Statistical Inference

Difference of Means Test

- $\bar{x}_1 - \bar{x}_2$
- We can do a permutation terst here too!

```
# calculate the observed statistic
observed_statistic <-  cb_tab_mf  %>%
  specify( ChipTime_min ~ Gender) %>%
  calculate(stat = "diff in means", order = c("F", "M"))

observed_statistic
```

```
## Response: ChipTime_min (numeric)
## Explanatory: Gender (factor)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  18.5
```

Statistical Inference

Difference of Means Test

- $\bar{x}_1 - \bar{x}_2$
- We can do a permutation terst here too!

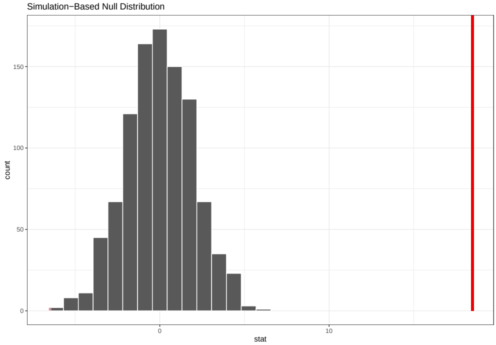
```
# generate the null distribution with randomization
null_dist_2_sample <- cb_tab_mf %>%
  specify(ChipTime_min ~ Gender) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("F", "M"))
```

Statistical Inference

Difference of Means Test

- $\bar{x}_1 - \bar{x}_2$
- We can do a permutation terst here too!

```
# generate the null distribution with randomization
null_dist_2_sample %>%
  visualize() +
  shade_p_value(observed_statistic,
                direction = "two-sided")+
  theme_bw()
```



Statistical Inference

Difference of Means Test

- $\bar{x}_1 - \bar{x}_2$
- We can do a permutation terst here too!

```
# generate the null distribution with randomization
## P-value
p_value_2_sample <- null_dist_2_sample %>%
  get_p_value(obs_stat = observed_statistic,
              direction = "two-sided")

p_value_2_sample
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

Statistical Inference

Difference of Means Test

- $\bar{x}_1 - \bar{x}_2$
- We can do a `t-test` here too
- $t = \bar{x}_1 - \bar{x}_2 / se(\bar{x}_1 - \bar{x}_2)$

```
## t-test
cb_tab_mf%>%t_test(
       formula = ChipTime_min ~ Gender,
       order = c("F", "M"),
       alternative = "two-sided")
```

```
## # A tibble: 1 x 7
##   statistic  t_df  p_value alternative estimate lower_ci upper_ci
##       <dbl> <dbl>    <dbl> <chr>          <dbl>    <dbl>    <dbl>
## 1      10.0 1186. 8.62e-23 two.sided       18.5     14.9     22.1
```

Difference of Means Test

- $\bar{x}_1 - \bar{x}_2$
- We can do a `t-test` here too
- $t = \bar{x}_1 - \bar{x}_2 / se(\bar{x}_1 - \bar{x}_2)$

```
# calculate the observed statistic
observed_statistic <- cb_tab_mf %>%
  specify(ChipTime_min ~ Gender,) %>%
  hypothesize(null = "point", mu = 0) %>%
  calculate(stat = "t", order = c("F", "M")) %>%
  dplyr::pull()

observed_statistic
```

```
##        t
## 10.03263
```

Statistical Inference

Difference of Means Test

- $\bar{x}_1 - \bar{x}_2$
- We can do a `t-test` here too
- $t = \bar{x}_1 - \bar{x}_2 / se(\bar{x}_1 - \bar{x}_2)$
- We can again compute the p-value
  - p-value < 0.05 - statistically significant (there is a difference!)
  - p-value > 0.05 - statistically not significant (there is **not** a difference!)

```
# calculate the observed statistic
observed_statistic <- cb_tab_mf %>%
  specify(ChipTime_min ~ Gender,) %>%
  hypothesize(null = "point", mu = 0) %>%
  calculate(stat = "t", order = c("F", "M")) %>%
  dplyr::pull()

observed_statistic
```

```
##        t
## 10.03263
```

```
pt(observed_statistic, df = nrow(gss) - 2, lower.tail = FALSE)*2
```

```
##            t
## 1.070199e-21
```

Statistical Inference

Difference of Means Test

- Old (over 35) versus Young (under 35)

```
# calculate the observed statistic
cb_tab_mf<-cb_tab_mf%>%mutate(
  youngOld = age_groups(Age,35),
  youngOld= recode_factor(youngOld, `0-34`="Young",
                          `35+`="Old")
)
levels(cb_tab_mf$youngOld)
```

```
## [1] "Young" "Old"
```

Statistical Inference

Difference of Means Test

- Old (over 35) versus Young (under 35)

```
observed_statistic <-  cb_tab_mf%>%filter(Gender=="M")  %>%
  specify(ChipTime_min ~ youngOld) %>%
  calculate(stat = "diff in means", order = c("Old","Young"))

observed_statistic
```

```
## Response: ChipTime_min (numeric)
## Explanatory: youngOld (factor)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  4.82
```

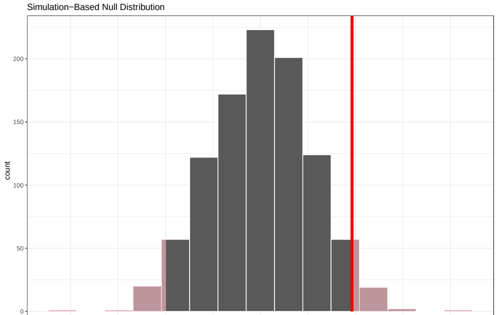
Statistical Inference

Difference of Means Test

- Old (over 35) versus Young (under 35)

```
# generate the null distribution with randomization
null_dist_2_sample <- cb_tab_mf%>%filter(Gender=="M") %>%
  specify(ChipTime_min ~ youngOld) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("Old","Young"))

null_dist_2_sample %>%
  visualize() +
  shade_p_value(observed_statistic,
                direction = "two-sided")+
  theme_bw()
```



Difference of Means Test

- Old (over 35) versus Young (under 35)

```
## P-value
p_value_2_sample <- null_dist_2_sample %>%
  get_p_value(obs_stat = observed_statistic,
              direction = "two-sided")

p_value_2_sample
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.068
```

Statistical Inference

Difference of Means Test

- Old (over 35) versus Young (under 35)

Statistical Inference

Difference of Means Test

- Old (over 35) versus Young (under 35)

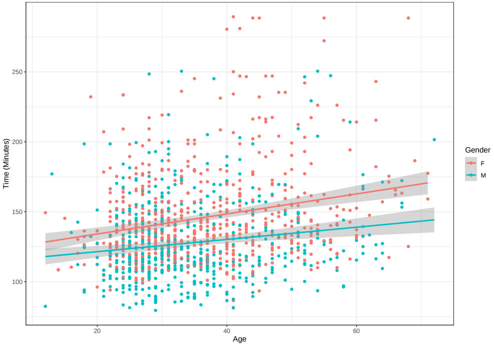
```
## t-test
cb_tab_mf%>%filter(Gender=="M")%>%t_test(
  formula = ChipTime_min ~ youngOld,
  order = c("Old","Young"),
  alternative = "two-sided")
```

```
## # A tibble: 1 x 7
##   statistic  t_df p_value alternative estimate lower_ci upper_ci
##       <dbl> <dbl>   <dbl> <chr>          <dbl>    <dbl>    <dbl>
## 1      1.82  422.  0.0687 two.sided       4.82   -0.372     10.0
```

Linear Regression

Linear Regression

```
cb_tab_mf%>%ggplot(aes(y=ChipTime_min,x=Age,color=Gender))+
  geom_point()+
  #stat_summary(fun.data= mean_cl_normal) +
  geom_smooth(method='lm')+
  theme_bw()+
  xlab("Age")+
  ylab("Time (Minutes)")
```



Linear Regression

- Simple Linear Regression

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

* Y in this case is run time in minutes (e.g. 140 minutes) * X is Age (e.g. 25) * Epsilon is deviation around the score (or measurment error) + $\epsilon \sim N(0, \sigma)$ (Normal Distribution) * R^2 is a measure of fit that ranges from 0 [low] to 1 [high]

```
summary(lm(ChipTime_min~Age,data=cb_tab_mf))
```

```
##
## Call:
## lm(formula = ChipTime_min ~ Age, data = cb_tab_mf)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -60.03 -21.61  -6.13  14.30 148.21
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 113.36433    3.18620  35.580  < 2e-16 ***
## Age           0.67918    0.08515   7.976 3.53e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.4 on 1186 degrees of freedom
## Multiple R-squared:  0.05091,    Adjusted R-squared:  0.05011
## F-statistic: 63.62 on 1 and 1186 DF,  p-value: 3.53e-15
```

Linear Regression

$$Y_i = \beta_0 + \beta_1 X_{1i} + ... + \beta_{ki} X_{ki} + \epsilon_{ki}$$

```
models<-list(
  lm(ChipTime_min~1,data=cb_tab_mf),
  lm(ChipTime_min~Age,data=cb_tab_mf),
lm(ChipTime_min~Age+Gender,data=cb_tab_mf)
)

r2<-models%>%purrr::map(function(x){summary(x)$r.squared})%>%unlist()
ar2<-models%>%purrr::map(function(x){summary(x)$adj.r.squared})%>%unlist()
data.frame(model=1:3,r2,ar2)
```

```
##   model         r2       ar2
## 1     1 0.00000000 0.0000000
## 2     2 0.05091324 0.0501130
## 3     3 0.11509494 0.1136014
```

Linear Regression

$$Y_i = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_{ki} X_{ki} + \epsilon_{ki}$$

```
models%>%purrr::map(function(x){broom::tidy(x)})%>%bind_rows(.id="models")
```

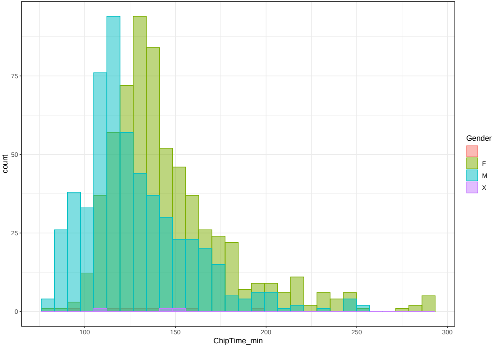
```
## # A tibble: 6 x 6
##   models term        estimate std.error statistic   p.value
##   <chr>  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 1      (Intercept)  138.        0.964     143.   0
## 2 2      (Intercept)  113.        3.19       35.6  2.90e-189
## 3 2      Age            0.679     0.0851      7.98 3.53e- 15
## 4 3      (Intercept)  124.        3.30       37.7  4.39e-205
## 5 3      Age            0.593     0.0828      7.16 1.42e- 12
## 6 3      GenderM      -17.0       1.83       -9.27 8.46e- 20
```

Visualizations

Visualization

- Histogram

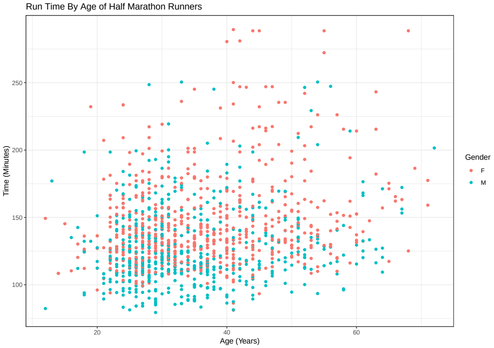
```
cb_tab%>%ggplot(aes(x=ChipTime_min, color=Gender,fill=Gender)) +
  geom_histogram(alpha=0.5, position="identity")+
  theme_bw()
```



Visualization

- Scatter Plot

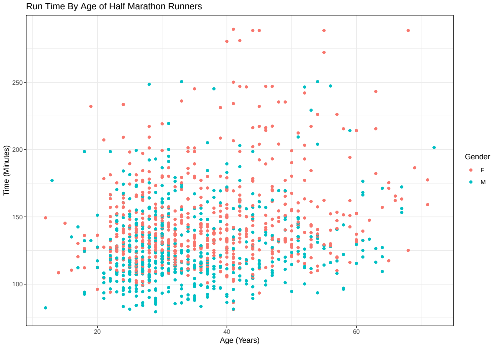
```
cb_tab_mf%>%ggplot(aes(y=ChipTime_min,x=Age,color=Gender))+
  geom_point()+
  theme_bw()+
  ggtitle("Run Time By Age of Half Marathon Runners")+
  xlab("Age (Years)")+
  ylab("Time (Minutes)")
```



Visualization

- Scatter Plot

```
cb_tab_mf%>%ggplot(aes(y=ChipTime_min,x=Age,color=Gender))+
  geom_point()+
  theme_bw()+
  ggtitle("Run Time By Age of Half Marathon Runners")+
  xlab("Age (Years)")+
  ylab("Time (Minutes)")
```

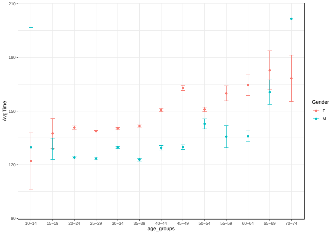


Visualization

- Line plot with SE

```
stats<-cb_tab_mf%>%group_by(age_groups,Gender)%>%summarise(
  Total=n(),
  AvgTime=mean(ChipTime_min),
  se=sd(ChipTime_min)/Total,
  med=median(ChipTime_min),
  q5=quantile(ChipTime_min,.05),
  q95=quantile(ChipTime_min,.95)
  )

stats%>%ggplot(aes(y=AvgTime,x=age_groups,group=Gender,color=Gender))+
  geom_point()+
  geom_errorbar(aes(ymin=AvgTime-2*se, ymax=AvgTime+2*se), width=.2)+
  ylim(95,205)+
  theme_bw()
```

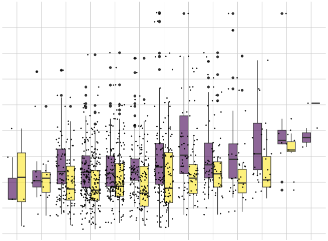


Visualization

- Box Plot

```
library(viridis)
library(hrbrthemes)

cb_tab_mf%>%
    ggplot( aes(x=age_groups, y=ChipTime_min, fill=Gender)) +
    geom_boxplot() +
    scale_fill_viridis(discrete = TRUE, alpha=0.6) +
    geom_jitter(color="black", size=0.4, alpha=0.9) +
    hrbrthemes::theme_ipsum() +
    theme(
        legend.position="none",
        plot.title = element_text(size=11)
    ) +
    ggtitle("Boxplot of Times by Age Groups") +
    xlab("Age Groups")+
    ylab("Time (Minutes)")
```



Summary

Basic Statistics in R for QS

- Descriptive statistics (mean, median, sd, se)
- Inferential statistics (permutation test, t-test, p-value)
- Linear Regression (Linear relationship between variables)

End of Slides