WorksheetWeek3

October 15, 2025

1 Data Transformation

This data set is from the same source as the College data in ISLR2. It has the US News data from ~1,300 US colleges, including admissions statistics, size, costs, and other aspects of the schools. ### Data Description

Most variables are named in quite obvious ways. To save some space, here are descriptions for variables that are potentially unclear: - FICE (Federal ID Number) - Public/private indicator (public=1, private=2) - Average Math SAT score - Average Verbal SAT score - Average Combined SAT score - Average ACT score - First quartile - Math SAT - Third quartile - Math SAT - First quartile - Verbal SAT - Third quartile - Verbal SAT - First quartile - ACT - Third quartile - ACT - Number of applications received - Number of applicants accepted - Number of new students enrolled - Pct. new students from top 10% of H.S. class - Pct. new students from top 25% of H.S. class - Pct. of faculty with Ph.D.'s - Pct. of faculty with terminal degree - Student/faculty ratio - Pct.alumni who donate - Instructional expenditure per student

[246]:

```
import pandas as pd
import altair as alt
import warnings
warnings.filterwarnings(
    "ignore",
    message=".*convert_dtype parameter is deprecated.*",
    category=FutureWarning
)
```

[248]:

```
dataurl = 'http://lib.stat.cmu.edu/datasets/colleges/usnews.data'
college = pd.read_csv(dataurl,
                      na_values='*',
                      names = ['FICE',
                               'College',
                               'State',
                               'Private',
                               'MathSAT',
                               'VerbalSAT',
                               'CombinedSAT',
                               'ACT',
                               'Q1Math',
                               'Q3Math',
                               'Q1Verbal',
                               'Q3Verbal',
                               'Q1ACT',
                               'Q3ACT',
                               'Applied',
                               'Accepted',
                               'Enrolled',
                               'Top10HS',
                               'Top25HS',
                               'FullTimeUG',
                               'PartTimeUG',
                               'InStateTuition',
                               'OutOfStateTuition',
                               'RoomBoardCost',
                               'RoomCost',
                               'BoardCost',
                               'ExtraFees',
                               'BookFees',
                               'PersonalFees',
                               'PhDFaculty',
                               'TerminalFaculty',
                               'StuFacRatio',
                               'AlumniDonate',
                               'SpendPerStudent',
                               'GradRate'])
```

1.0.1 Plot 1

Make a scatterplot (called plot1) of the in-state tuition costs vs out-of-state tuition for public institutions. Change the aspect ratio and domain of the plot so it shows the data most clearly, clipping outliers. Using `clip=True` as a parameter in `mark_*()` may be helpful.

Above it, plot a bar chart histogram of in-state tuition for public institutions (called plot2).

To the right, add another horizontal-bar histogram of out-of-state tuition for public institutions, each binned by $1000 increments (called plot3).

Make sure the sizes of the plots correspond (e.g. your top histogram should be the same x-axis length as your scatter plot x-axis) and all plots displayed within the page.

[250]:

```
plot1 = alt.Chart(college).mark_circle(clip=True).encode(
    alt.X('InStateTuition:Q'),
    alt.Y('OutOfStateTuition:Q')
).transform_filter('datum.Private == "1"')

plot2 = alt.Chart(college).mark_bar().encode(
    alt.X('InStateTuition:Q', bin=alt.BinParams(maxbins=20)),
    alt.Y('count()')
).transform_filter('datum.Private == "1"')
```

```
plot3= alt.Chart(college).mark_bar().encode(
    alt.X('count()'),
    alt.Y('OutOfStateTuition:Q', bin=alt.BinParams(maxbins=20))
).transform_filter('datum.Private == "1"')
plot2 & ( plot1 | plot3 ) #shows the position of the plot displayed
```

```
[250]: alt.VConcatChart(…)
```

1.0.2 Plot 2

Create a new variable called AcceptRate to compute the acceptance rate based on the number of applications and acceptances.

Make a scatterplot of the percent of students in the top 10% of their high school class vs the acceptance rate. Color the points by the expenditure per student, then rescale the colorbar by including `scale=alt.Scale(type='log', scheme = 'redyellowgreen', nice = True` so the variation in expenditure is more clear.

```
[256]: plot1 = alt.Chart(college).mark_circle(clip=True).transform_calculate(
           AcceptRate='datum.Accepted / datum.Applied'
       ).encode(
           alt.X('Top10HS:Q'),
           alt.Y('AcceptRate:Q'),
           alt.Color('SpendPerStudent:Q',scale=alt.Scale(type='log', scheme =␣
         ↪'redyellowgreen', nice = True))
       )
       plot1
```

```
[256]: alt.Chart(…)
```

1.0.3 Plot 3

Calculate a new variable called Revenue, estimated using the in-state tuition, room and board costs, and full-time undergraduate population.

Make a heatmap where, by state, the total revenue is shown for different school sizes (binned). Filter out null values.

What question can this plot answer?

```
[259]: import altair as alt
       import pandas as pd

       heatmap = (
           alt.Chart(college)
           .transform_filter(
               "(datum.State != null) && "
               "(datum.InStateTuition != null) && "
               "(datum.RoomCost != null) && "
               "(datum.BoardCost != null) &&"
```

```
        "(datum.FullTimeUG != null)"
    )
    .transform_calculate(
        Revenue="((datum.InStateTuition) + (datum.RoomCost) + (datum.
 ↪BoardCost)) * (datum.FullTimeUG)"
    )
    .mark_bar()
    .encode(
        x=alt.X("State:N",sort=alt.Sort(field="sum(Revenue)",␣
 ↪order="descending"),title="State"),
        y=alt.Y("FullTimeUG:Q", bin=alt.Bin(step=5000),title="School size"),
        color=alt.Color("sum(Revenue):Q",title="Total revenue",scale=alt.
 ↪Scale(scheme="blues")),
        tooltip=[
            alt.Tooltip("State:N"),
            alt.Tooltip("FullTimeUG:Q", title="Size bin (upper)"),
            alt.Tooltip("sum(Revenue):Q", title="Total revenue")
        ],
    )
    .properties(width=900, height=420, title="Estimated Total Revenue by State␣
 ↪and School Size")
)
heatmap
```

[259]: alt.Chart(…)

This plot helps identify which states generate the highest total estimated revenue from tuition and room & board, and whether that revenue mainly comes from large or small schools. It reveals how the financial scale of higher education systems varies across states.

1.0.4 Plot 4

Make a scatterplot the graduation rate vs student to faculty ratio (specifying `scale` may be useful here) for public and private. Note that we have a graduation rate higher than 100%… Change the domain so we only see reasonable values ranging 0-100 (using `clip=True` as a parameter in `mark_*()` may be helpful here). Color the points by public/private and choose two new colors from the default and the examples from class.

Then, beside it, make the same plot with a new goal: showing Seattle U, encoding with both size and color. Make all of the points a consistent circle of light gray color (both public and private) except for Seattle U, which should be large and a salient color from the official SU branding guidelines.

Place the legend for each above the respective plots. Make sure to label your axes and legend clearly.

[263]:
```
color1 = (
    alt.Chart(college)
    .transform_filter(
        "(datum.GradRate != null) && (datum.StuFacRatio != null)"
    )
    .transform_calculate(
        PublicPrivate="datum.Private === 2 ? 'Private' : 'Public'"
    )
    .mark_circle(clip=True)
    .encode(
        x=alt.X(
            "StuFacRatio:Q",
            title="Student-to-Faculty Ratio"
        ),
        y=alt.Y(
            "GradRate:Q",
            scale=alt.Scale(domain=[0, 100], nice=False),
            title="Graduation Rate (%)"
        ),
        color=alt.Color(
            "PublicPrivate:N",
            scale=alt.Scale(domain=["Public","Private"], range=["#4169e1",␣
 ↪"#cd5c5c"]),
            title="Institution Type",
            legend=alt.Legend(orient="top")
        ),
        tooltip=[
            "College:N", "State:N", "PublicPrivate:N",
            alt.Tooltip("GradRate:Q", title="Grad Rate (%)"),
            alt.Tooltip("StuFacRatio:Q", title="Stu-Fac Ratio")
        ],
        size=alt.value(60)
    )
    .properties(
        width=380,
        height=320,
        title="Graduation Rate vs Student-Faculty Ratio (Public vs Private)"
    )
)
```

[265]:

```
color2 = (
    alt.Chart(college)
    .transform_filter(
        "(datum.GradRate != null) && (datum.StuFacRatio != null)"
    )
    .transform_calculate(
        PublicPrivate="datum.Private === 2 ? 'Private' : 'Public'",
        Highlight="datum.College === 'Seattle University' ? 'Seattle␣
 ↪University' : 'Other Schools'"
    )
    .mark_circle(clip=True)
    .encode(
        x=alt.X(
            "StuFacRatio:Q",
            title="Student-to-Faculty Ratio"
        ),
        y=alt.Y(
            "GradRate:Q",
            scale=alt.Scale(domain=[0, 100], nice=False),
            title="Graduation Rate (%)"
        ),
        color=alt.Color(
            "Highlight:N",
            scale=alt.Scale(domain=["Other Schools","Seattle University"],␣
 ↪range=["#D3D3D3", "#AA0000"]),
            title="Highlight",
            legend=alt.Legend(orient="top")
        ),
        size=alt.condition(
            alt.datum.Highlight == "Seattle University",
            alt.value(220),
            alt.value(60)
        ),
        tooltip=[
            "College:N", "State:N",
            alt.Tooltip("GradRate:Q", title="Grad Rate (%)"),
            alt.Tooltip("StuFacRatio:Q", title="Stu-Fac Ratio")
        ],
    )
    .properties(
        width=380,
        height=320,
        title="Seattle University"
    )
)
```

[267]:

```
alt.hconcat(color1, color2).resolve_scale(color='independent')
```

[267]:

```
alt.HConcatChart(…)
```

[ ]: